



Universidad Científica del Perú - UCP
*Registrado en el Asiento N° A00010 de la Partida N° 11000318, Personas Jurídicas de Iquitos,
Superintendencia de los Registros Públicos - SUNARP*

FACULTAD DE CIENCIAS E INGENIERIA
PROGRAMA ACADÉMICO DE INGENIERIA DE SISTEMAS DE
INFORMACION

TESIS

INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE
SOFTWARE EN LA CALIDAD DE LAS APLICACIONES
PRODUCIDAS POR LA EMPRESA “SOFTWARE Y SISTEMAS DEL
PERÚ S.A.C., IQUITOS - 2021”

PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS DE INFORMACIÓN

AUTOR : Br. MÁRQUEZ NAVARRO, George Patrick
Br. LINARES CHUMBE, Saulo Patrick

ASESOR : Ing. MARTHANS RUÍZ, Ángel Alberto


.....
Ángel Alberto Marthans Ruiz
Ingeniero de Sistemas
Reg. CIP N° 219877

San Juan Bautista - Loreto – Maynas – 2022

Dedicatoria

El presente trabajo va dedicado a mi madre por sus enseñanzas, apoyo moral e incondicional; a mi esposa y a mis dos hijos siendo el motor y motivo de seguir siempre adelante y por todo el apoyo brindado durante mi formación personal y profesional.

Br. Saulo Patrick Linares Chumbe.

Dedicatoria

A Dios por enviarme a esta vida a ser algún engranaje para sus propósitos, a mi madre Yrma que gracias a su fortaleza sus oraciones y sus enseñanzas han sido un faro en mi camino para convertirme en un profesional con éxito y a mi papá Carlos Quiroz por ser un pilar importante para cimentar las bases de mi vida entera.

Br. George Patrick Márquez Navarro.

Agradecimiento

Expresamos nuestra gratitud y agradecimiento a Dios sobre todas las cosas por brindarnos la salud y la de nuestros seres queridos, a nuestros padres por habernos guiado por el buen camino, haciendo posible nuestra formación profesional, personal y espiritual, a nuestro asesor y docentes gracias por brindarnos su tiempo dedicación y empeño en compartir sus conocimientos, por último y no menos importante a la Empresa Software y Sistemas del Perú SAC por brindarnos todas las facilidades de hacer posible este trabajo de investigación.

**CONSTANCIA DE ORIGINALIDAD DEL TRABAJO DE INVESTIGACIÓN
DE LA UNIVERSIDAD CIENTÍFICA DEL PERÚ - UCP**

El presidente del Comité de Ética de la Universidad Científica del Perú - UCP

Hace constar que:

La Tesis titulada:

**"INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE
EN LA CALIDAD DE LAS APLICACIONES PRODUCIDAS POR LA EMPRESA
"SOFTWARE Y SISTEMAS DEL PERÚ S.A.C., IQUITOS - 2021"**

De los alumnos: **MARQUEZ NAVARRO GEORGE PATRICK Y LINARES
CHUMBE SAULO PATRICK**, de la Facultad de Ciencias e
Ingeniería, pasó satisfactoriamente la revisión por el Software Antiplagio,
con un porcentaje de **8% de plagio**.

Se expide la presente, a solicitud de la parte interesada para los fines que
estime conveniente.

San Juan, 12 de Agosto del 2022.



Dr. César J. Ramal Asayag
Presidente del Comité de Ética – UCP

Document Information

Analyzed document	UCP_INGENIERIA_2022_TESIS_LUISMARQUEZ_OSCARLINARES_V1.pdf (D142660337)
Submitted	2022-08-04 18:26:00
Submitted by	Comisión Antiplagio
Submitter email	revision.antiplagio@ucp.edu.pe
Similarity	8%
Analysis address	revision.antiplagio.ucp@analysis.urkund.com

Sources included in the report

SA	TESIS WENDY MARCILLO LIGUA.pdf Document TESIS WENDY MARCILLO LIGUA.pdf (D108962908)	 8
SA	PROYECTO_TESIS_WENDY_MARCILLO1.docx Document PROYECTO_TESIS_WENDY_MARCILLO1.docx (D108212615)	 1
SA	TESIS7.pdf Document TESIS7.pdf (D44006917)	 1
SA	Marcillo Ligua.pdf Document Marcillo Ligua.pdf (D109004535)	 1

Entire Document

FACULTAD DE CIENCIAS E INGENIERIA PROGRAMA ACADÉMICO DE INGENIERIA DE SISTEMAS DE INFORMACION
 TESIS INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE EN LA CALIDAD DE LAS
 APLICACIONES PRODUCIDAS POR LA EMPRESA "SOFTWARE Y SISTEMAS DEL PERÚ S.A.C., IQUITOS - 2021" PARA
 OPTAR EL TÍTULO PROFESIONAL EN CIENCIAS E INGENIERÍA. MENCIÓN: INGENIERÍA DE INFORMÁTICA Y DE
 SISTEMAS AUTOR : Br. MARQUEZ NAVARRO, Luis Fernando Br. LINARES CHUMBE, Oscar David ASESOR : Ing.
 MARTHANS RUIZ, Ángel Alberto San Juan Bautista - Loreto – Maynas – 2022
 UCP-logoweb-2017-3

2 Dedicatoria El presente trabajo va dedicado a mi madre por sus enseñanzas, apoyo moral e incondicional; a mi esposa y a mis dos hijos siendo el motor y motivo de seguir siempre adelante y por todo el apoyo brindado durante mi formación personal y profesional. Br. Saulo Patrick Linares Chumbe. Dedicatoria A Dios por enviarme a esta vida a ser algún engranaje para sus propósitos, a mi madre Yrma que gracias a su fortaleza sus oraciones y sus enseñanzas han sido un faro en mi camino para convertirme en un profesional con éxito y a mi papá Carlos Quiroz por ser un pilar importante para cimentar las bases de mi vida entera. Br. George Patrick Márquez Navarro.

3 Agradecimiento Expresamos nuestra gratitud y agradecimiento a Dios sobre todas las cosas por brindarnos la salud y la de nuestros seres queridos, a nuestros padres por habernos guiado por el buen camino, haciendo posible nuestra formación profesional, personal y espiritual, a nuestro asesor y docentes gracias por brindarnos su tiempo dedicación y empeño en compartir sus conocimientos, por último y no menos importante a la Empresa Software y Sistemas del Perú SAC por brindarnos todas las facilidades de hacer posible este trabajo de investigación.

"Año del Fortalecimiento de la Soberanía Nacional"

ACTA DE SUSTENTACIÓN DE TESIS

FACULTAD DE CIENCIAS E INGENIERÍA

Con Resolución Decanal N° 896-2021-UCP-FCEI del 07 de diciembre del 2021, la FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD CIENTÍFICA DEL PERÚ - UCP designa como Jurado Evaluador de la sustentación de tesis a los señores:

- | | |
|--|------------|
| • Ing. Jimmy Max Ramírez Villacorta, Mgr. | Presidente |
| • Ing. Tonny Eduarco Bardales Lozano, Mgr. | Miembro |
| • Lic. Carlos Enrique Marthans Ruiz, Mgr. | Miembro |

Como Asesor: Ing. Angel Alberto Marthans Ruiz, Mg.

En la ciudad de Iquitos, siendo las 09:00:00 horas del día 20 de setiembre del 2022, de manera Virtual, empleando la plataforma ZOOM y supervisado por la Secretaría Académica del programa Académico de Ingeniería de Sistemas de Información de la Facultad de Ciencias e Ingeniería de la Universidad Científica del Perú, se constituyó el Jurado para escuchar la sustentación y defensa de la Tesis: "INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE EN LA CALIDAD DE LAS APLICACIONES PRODUCIDAS POR LA EMPRESA "SOFTWARE Y SISTEMAS DEL PERU S.A.C., IQUITOS-2021"

Presentado por los sustentantes: **GEORGE PATRICK MARQUEZ NAVARRO Y SAULO PATRICK LINARES CHUMBE**

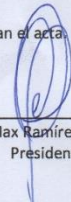
Como requisito para optar el título profesional de: **INGENIERO DE SISTEMAS DE INFORMACIÓN**

Luego de escuchar la sustentación y formuladas las preguntas las que fueron: **ABSUELTAS**

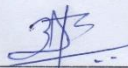
El Jurado después de la deliberación en privado llegó a la siguiente conclusión:

La sustentación es: **APROBADO**

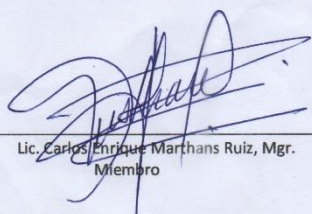
En fe de lo cual los miembros del Jurado firman el acta



Ing. Jimmy Max Ramírez Villacorta, Mgr
Presidente



Ing. Tonny Eduarco Bardales Lozano, Mgr.
Miembro



Lic. Carlos Enrique Marthans Ruiz, Mgr.
Miembro

Contáctanos: **Iquitos - Perú** 065 - 26 1088 / 065 - 26 2240 Av. Abelardo Quiñones Km. 2.5 **Filial Tarapoto - Perú** 42 - 58 5638 / 42 - 58 5640 Leoncio Prado 1070 / Martines de Compagñon 933 Universidad Científica del Perú www.ucp.edu.pe

Índice de contenido

Dedicatoria	2
Agradecimiento	3
Páginas de aprobación	4
Resumen	11
Abstract	12
Introducción	13
Capítulo I. Marco teórico	14
1.1 Antecedentes del estudio	14
1.1.1 Internacionales	14
1.1.2 Nacionales	14
1.1.3 Locales	16
1.2 Bases teóricas	16
1.2.1 Metodología de Desarrollo de Software	16
1.2.2 Metodologías Ágiles	20
1.2.3 Metodologías basadas en modelos	27
1.2.4 Calidad de Software	28
1.3 Definición de términos básicos	36
Capítulo II. Planteamiento del problema	38
2.1 Descripción del problema	38
2.2 Formulación del problema	39
2.2.1 Problema general	39
2.2.2 Problemas específicos	39
2.3 Objetivos	40
2.3.1 Objetivo general	40
2.3.2 Objetivos específicos	40
2.4 Justificación de la investigación	41
2.5 Hipótesis	42
2.5.1 Hipótesis General	42
2.5.2 Hipótesis específicas	42
2.6 Variables	43
2.6.1 Identificación de variables	43
2.6.2 Definición de las variables	43
2.6.3 Operacionalización de las variables	44
Capítulo III. Metodología	45
3.1 Tipo y diseño de investigación	45
3.1.1 Tipo de investigación	45
3.1.2 Diseño de investigación	45
3.2 Población y muestra	46
3.2.1 Población	46
3.2.2 Muestra	46

3.2.3	Participantes	47
3.3	Técnicas, instrumentos y procedimientos de recolección de datos	48
3.3.1	Técnicas de recolección de datos	48
3.3.2	Instrumentos de recolección de datos	48
3.3.3	Procedimientos de recolección de datos	49
3.4	Procesamiento y análisis de datos	49
3.4.1	Procesamiento de los datos	49
3.4.2	Análisis de los datos	49
	Capítulo IV. Resultados	50
4.1	Resultados	50
4.1.1	Estadística Descriptiva	50
4.1.2	Análisis de los resultados de la variable independiente – Metodología de Desarrollo de Software	53
4.1.3	Análisis de los resultados de la variable dependiente – Calidad de Software	59
	Capítulo V. Discusión, conclusiones y recomendaciones	65
5.1	Discusión	65
5.2	Conclusiones	68
5.3	Recomendaciones	69
	Referencias bibliográficas	70
	Anexos	72
	Anexo 01: Matriz de consistencia	73
	Anexo 02: Instrumento de recolección de datos	74

Índice de cuadros o tablas

Tabla 01: Operacionalización de variables	44
Tabla 02: Trabajadores encuestados	47
Tabla 03: Estadística descriptiva para el índice Muy Alto (5)	50
Tabla 04: Estadística descriptiva para el índice Alto (4)	50
Tabla 05: Estadística descriptiva para el índice Medio (3)	51
Tabla 06: Estadística descriptiva para el índice Bajo (2)	51
Tabla 07: Estadística descriptiva para el índice Muy Bajo (1)	52
Tabla 08: Efectividad - Nivel de comunicación entre los interesados del proyecto	53
Tabla 09: Efectividad - Nivel de capacidad de autogestión del equipo de trabajo	54
Tabla 10: Costo - Nivel de costo de conformidad	55
Tabla 11: Costo - Nivel de costo de no conformidad	56
Tabla 12: Flexibilidad - Nivel de respuesta a cambios en los procesos de la organización	57
Tabla 13: Flexibilidad - Nivel de adaptabilidad del software a los requisitos cambiantes del proyecto	58
Tabla 14: Usabilidad - Nivel de facilidad en el aprendizaje del uso del software	59
Tabla 15: Usabilidad - Nivel de facilidad en el uso operativo del software	60
Tabla 16: Fiabilidad - Nivel de tolerancia a fallos del software	61
Tabla 17: Fiabilidad - Nivel de exactitud en el procesamiento de la información	62
Tabla 18: Mantenibilidad - Nivel de facilidad para localizar un fallo operativo del software	63
Tabla 19: Mantenibilidad - Nivel de facilidad para arreglar un fallo operativo del software	64

Índice de gráficos o figuras

Gráfico 01: Efectividad - Nivel de comunicación entre los interesados del proyecto	53
Gráfico 02: Efectividad - Nivel de capacidad de autogestión del equipo de trabajo	54
Gráfico 03: Costo - Nivel de costo de conformidad	55
Gráfico 04: Costo - Nivel de costo de no conformidad	56
Gráfico 05: Flexibilidad - Nivel de respuesta a cambios en los procesos de la organización	57
Gráfico 06: Flexibilidad - Nivel de adaptabilidad del software a los requisitos cambiantes del proyecto	58
Gráfico 07: Usabilidad - Nivel de facilidad en el aprendizaje del uso del software	59
Gráfico 08: Usabilidad - Nivel de facilidad en el uso operativo del software	60
Gráfico 09: Fiabilidad - Nivel de tolerancia a fallos del software	61
Gráfico 10: Fiabilidad - Nivel de exactitud en el procesamiento de la información	62
Gráfico 11: Mantenibilidad - Nivel de facilidad para localizar un fallo operativo del software	63
Gráfico 12: Mantenibilidad - Nivel de facilidad para arreglar un fallo operativo del software	64

RESUMEN

El presente trabajo de investigación, que lleva como título: “INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE EN LA CALIDAD DE LAS APLICACIONES PRODUCIDAS POR LA EMPRESA “SOFTWARE Y SISTEMAS DEL PERÚ S.A.C.”, IQUITOS - 2021”, tuvo como objetivo general describir cómo el uso de metodologías de desarrollo de software, influye en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.” de la ciudad de Iquitos – 2021.

Para el desarrollo de la investigación, se utilizó un enfoque descriptivo, realizándose un análisis estadístico a través de la recolección de datos, que sirvieron para comprobar la hipótesis y proponer una solución. El tipo de investigación fue descriptiva, para establecer la relación entre las variables *Metodología de Desarrollo de Software* y *Calidad de Software*. El diseño de la investigación, fue no experimental transversal de tipo correlacional, pues se aplicaron encuestas en un único momento del desarrollo de la investigación. La población fue de 11 trabajadores de la empresa Software y Sistemas del Perú S.A.C., de la ciudad de Iquitos, correspondiendo la muestra del estudio, a la totalidad de la población. Se aplicó la técnica de la encuesta, a través de un cuestionario de 24 preguntas, para cuyo análisis e interpretación, se utilizó la escala de Likert.

Esta investigación y su respectivo análisis de datos, comprueban que el uso de metodologías de desarrollo de software, incrementan el nivel de calidad de las aplicaciones producidas por la empresa Software y Sistemas del Perú S.A.C – Iquitos; repercutiendo, asimismo, en una eficiente productividad laboral y en la optimización del servicio al cliente.

PALABRAS CLAVE: Metodología de Desarrollo de Software, Software, Desarrollo de Software, Calidad, Reducción de costos, Organización, Eficiencia, Productividad.

ABSTRACT

The present research work, entitled: "INFLUENCE OF THE USE OF SOFTWARE DEVELOPMENT METHODOLOGIES ON THE QUALITY OF THE APPLICATIONS PRODUCED BY THE COMPANY "SOFTWARE Y SISTEMAS DEL PERÚ S.A.C. IQUITOS - 2021", had as general objective to describe how the use of software development methodologies, influences the quality of the applications produced by the company "Software y Sistemas del Perú S.A.C." of the city of Iquitos - 2021.

This research used a descriptive approach, performing a statistical analysis through data collection, which served to test the hypothesis and propose a solution. The type of research was descriptive, to find the relationship between the variables Software Development Methodology and Software Quality. The research design was non-experimental, cross-sectional and correlational, since surveys were applied at a single time during the development of the research. The population consisted of 11 workers of the company *Software y Sistemas del Perú S.A.C.*, in the city of Iquitos, corresponding to the entire population of the study sample. The survey technique was applied through a questionnaire of 24 questions, for whose analysis and interpretation the Likert scale was used.

The research and analysis of the data, conclude that the use of software development methodologies generates a positive impact on the quality of the applications produced by the company *Software y Sistemas del Perú S.A.C - Iquitos*, also affecting labor productivity and the quality of customer service.

KEY WORDS: Software Development Methodology, Software, Software Development, Quality, Cost Reduction, Organization, Efficiency, Productivity.

INTRODUCCIÓN

Toda organización o proyecto, requiere un orden para su funcionamiento. El campo de la Administración, nos ha legado pautas y principios tendientes a alcanzar este orden y a lograr eficiencia. En el mundo empresarial, se siguen estos principios administrativos que garantizan un funcionamiento óptimo, maximizando el uso de los recursos y reduciendo riesgos y contingencias.

Esta herencia administrativa, ha calado en la gestión de todo proyecto. En el campo de la tecnología e informática, se utilizan, ahora, diferentes técnicas y procedimientos, para alcanzar ese orden y eficiencia.

Y, en la actualidad, donde nos encontramos viviendo la época de la TRANSFORMACIÓN DIGITAL, las empresas reconfiguran sus estructuras y reestructuran sus procesos, ya no sólo para alcanzar un orden, sino para ser cada vez más competitivas. De igual forma, los clientes o usuarios de estas empresas, muestran características cada vez más complejas y exigentes. Para responder a estas características, las empresas deben desarrollar estrategias que le permitan estar al nivel de esas exigencias, debiendo ofrecerle a clientes y usuarios, soluciones tecnológicas rápidas, eficientes y sostenibles en el tiempo.

Estas soluciones tecnológicas, que inspiran la presente investigación, pueden tratarse de aplicaciones y sistemas, que deben desarrollarse siguiendo procedimientos que faciliten su construcción, para obtener productos de alta calidad. La búsqueda de orden, eficiencia y competitividad, se ponen nuevamente de manifiesto, en cada aspecto del accionar humano.

Estos procedimientos, se traducen en metodologías que organizan el trabajo de la manera más ordenada posible y que buscan desarrollar aplicaciones con óptimos estándares de calidad, cumpliendo los requerimientos y necesidades del cliente/usuario.

A mayor organización del trabajo de producción de soluciones tecnológicas, mejor será su calidad y mayor será la satisfacción del cliente que lo solicite.

CAPÍTULO I

MARCO TEÓRICO

1.1. ANTECEDENTES DEL ESTUDIO

1.1.1 Antecedentes Internacionales

(QUEZADA MAURICIO, 2014), en su investigación titulada “BUENAS PRÁCTICAS DENTRO DEL MARCO DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE, PARA EL USO DE LOS ESTUDIANTES DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”, concluye que la aplicación de una metodología de desarrollo de software, es de gran ayuda para paliar las debilidades encontradas durante el diagnóstico, debilidades en la administración de proyectos, la baja calidad de los productos de software, la falta de automatización en procesos de control y seguimiento, así como en la administración y la integración de módulos. Las buenas prácticas, colaboran con la mejora de la estimación de recursos, a lo largo de los proyectos de desarrollo de software.

1.1.2 Antecedentes nacionales

(ARAUJO COTACALLAPA, 2017), en su investigación titulada, “SISTEMATIZACIÓN DE LOS PROCESOS DEL PLAN OPERATIVO INSTITUCIONAL UTILIZANDO LA METODOLOGÍA PUDS, PARA LA ELABORACIÓN DEL SISTEMA DE INFORMACIÓN PLAN OPERATIVO INSTITUCIONAL (POIS) EN LA UNIVERSIDAD NACIONAL DEL ALTIPLANO”, concluye que, con el Sistema de Información Plan Operativo Institución al, el acceso a la información detallada es instantánea y está dado por acciones estratégicas. El almacenamiento y la consulta de

información en el Plan Operativo Institucional, es eficiente y posible, permitiendo acceder a la información en forma oportuna (tiempo real) para los responsables de los Centros de Costo (directores de Escuela de Unidad Académica, jefes de oficina y unidad de áreas administrativas). Así también, se puede realizar el seguimiento y monitoreo de las acciones estratégicas programadas.

(HUAMAN VARAS, y otros, 2017), en su investigación titulada, “DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA MEJORAR LOS PROCESOS DE COMPRAS Y VENTAS EN LA EMPRESA HUMAJU”, se comprueba que la metodología Ágil AUP, permitió conocer los riesgos de gestión, en donde estos tengan prioridad en el proceso de desarrollo del sistema, el cual busca darle una mayor satisfacción al cliente, mediante la entrega temprana del software. Se pudo observar que esta metodología, se centra en la funcionalidad que el sistema debe poseer, para satisfacer las necesidades del usuario y siendo una metodología con claridad en sus actividades, en las etapas del diseño y construcción de software.

(GODOY ALVAREZ, y otros, 2015), en su investigación titulada, “MODELOS DE ACEPTACIÓN DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE”, se concluye que, de los 17 factores que se establecieron como impulsores en la aceptación de las metodologías, los que fueron estudiados más veces en los modelos, son: Compatibilidad, Utilidad, Facilidad de condiciones y Voluntad de uso. Finalmente, los jefes de proyectos de desarrollo, deben tomar en consideración los factores impulsores e inhibidores establecidos en esta investigación, para que el proceso de aceptación de metodologías de desarrollo de software, sea el más adecuado.

1.1.3 Antecedentes locales

(PEZO AREVALO, 2020), en su investigación titulada “IMPLEMENTACIÓN DE SISTEMA DE CONTROL DE ASISTENCIA DOCENTE, PARA MEJORAR LA GESTIÓN DE PERSONAL ACADÉMICO EN LA INSTITUCIÓN EDUCATIVA PÚBLICA DE MENORES N° 60023 - IQUITOS”, concluye que, en relación a su hipótesis general, la implementación del sistema de control de asistencia docente, tuvo efectos de mejora en la gestión de personal académico.

1.2 BASES TEÓRICAS

1.2.1. Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software, son un conjunto de técnicas y métodos organizativos, que se aplican para diseñar soluciones de software informático. El objetivo de estas metodologías, es el de organizar equipos de trabajo, para que estos desarrollen las funciones de un programa de la mejor manera posible. (Santander Universidades, 2020).

Cuando se trata de desarrollar productos o soluciones para un cliente o mercado concreto, es necesario tener en cuenta factores como los costos, la planificación, la dificultad, el equipo de trabajo disponible, los lenguajes utilizados, etc. Todos ellos, se engloban en una metodología de desarrollo, que permite organizar el trabajo de la forma más ordenada posible.

El desarrollo de software, puede ser un sector especialmente complejo, sobre todo cuando se trata de grandes aplicativos y equipos de trabajo. Ponerse a desarrollar un producto sin una metodología clara, desembocará en un proceso aún más complejo,

que conducirá a problemas, retrasos, errores y, en definitiva, a un mal resultado final.

El trabajo con una metodología de desarrollo de software, permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar. En la actualidad, se pueden diferenciar dos grandes grupos de metodologías de desarrollo de software: las ágiles y las tradicionales.

A continuación, se explican las características de cada una de ellas.

- **Metodologías de desarrollo de software tradicionales:**

Las metodologías de desarrollo de software tradicionales, se caracterizan por definir total y rígidamente, los requisitos al inicio de los proyectos de ingeniería de software. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, al contrario de las metodologías ágiles, lo que ha propiciado el incremento en el uso de las segundas.

La organización del trabajo de las metodologías tradicionales, es lineal; es decir, las etapas se suceden una tras otra y no se puede empezar la siguiente, sin terminar la anterior. Tampoco se puede volver hacia atrás, una vez se ha cambiado de etapa. Estas metodologías, no se adaptan nada bien a los cambios y el mundo actual cambia constantemente. Las principales metodologías tradicionales o clásicas, son:

- **Waterfall (cascada):** Es una metodología en la que las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa, se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales, no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.

- **Prototipado:** Se basa en la construcción de un prototipo de software que se construye rápidamente, para que los usuarios puedan probarlo y aportar feedback¹. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir. Es un modelo iterativo, que se basa en el método de prueba y error, para comprender las especificidades del producto.
- **Espiral:** Es una combinación de los dos modelos anteriores, que añade el concepto de análisis de riesgo. Se divide en cuatro etapas: planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. El nombre de esta metodología, se debe a su funcionamiento, ya que se van procesando las etapas en forma de espiral. Cuanto más cerca se está del centro, más avanzado está el proyecto.
- **Incremental:** En esta metodología de desarrollo de software, se va construyendo el producto final de manera progresiva. En cada etapa incremental, se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El software se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es mucho más flexible que las demás metodologías.
- **Diseño rápido de aplicaciones (RAD):** Esta metodología, permite desarrollar software de alta calidad en un corto periodo de tiempo. Los costes son mucho más altos y el desarrollo, más flexible, aunque requiere una mayor intervención de los usuarios. Por otro lado, el código puede contener más errores y sus funciones son limitadas, debido al poco tiempo del que se dispone para desarrollarlas. El objetivo es iterar el menor número posible de veces, para conseguir una aplicación completa de forma rápida.

¹ El feedback o retroalimentación, es un mecanismo por el cual una cierta proporción de la salida de un sistema se dirige a la entrada, con señales de controlar su comportamiento. Es un proceso administrativo donde el control es una etapa cualitativa y cuantitativa, que sirve de base para la fase de planeación. <https://es.wikipedia.org/wiki/Realimentaci%C3%B3n> – Consulta realizada el 11/11/2021.

- **Metodologías de desarrollo de software ágiles:**

Las metodologías ágiles de desarrollo de software, son las más utilizadas hoy en día, debido a su alta flexibilidad y agilidad. Los equipos de trabajo que las utilizan, son mucho más productivos y eficientes, pues saben lo que tienen que hacer en cada momento. Además, estas metodologías permiten adaptar el software a las necesidades que van surgiendo por el camino, lo que facilita construir aplicaciones más funcionales. Las metodologías ágiles se basan en la metodología incremental, en la que, en cada ciclo de desarrollo, se van agregando nuevas funcionalidades a la aplicación final. Sin embargo, los ciclos son mucho más cortos y rápidos, por lo que se van agregando pequeñas funcionalidades en lugar de grandes cambios. Este tipo de metodologías, permiten construir equipos de trabajo autosuficientes e independientes que se reúnen cada poco tiempo, para poner en común las novedades. Poco a poco, se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones, ya que puede comprobar cómo avanza el proyecto en tiempo real. Las principales metodologías ágiles son:

- **Kanban:** Es una metodología de trabajo, inventada por la empresa de automóviles Toyota. Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo, dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual, basado en tareas prioritarias e incrementando el valor del producto.
- **Scrum:** Es también una metodología incremental, que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas), para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (*planning sprint*), ejecución (*sprint*), reunión diaria (*daily meeting*) y demostración de resultados (*sprint review*). Cada iteración por estas etapas, se denomina también *sprint*.

- **Lean:** Está configurado para que pequeños equipos de desarrollo muy capacitados, elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, relegando a un segundo plano, el tiempo y los costes. El aprendizaje, las reacciones rápidas y potenciar el equipo, son fundamentales.
- **Programación extrema (XP):** Es una metodología de desarrollo de software basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo, es crear un buen ambiente de trabajo en equipo y que haya un *feedback* constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, *testing*², refactorización³ y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua⁴, entregas semanales e integridad con el cliente, cliente *in situ*, entregas frecuentes y planificación.

1.2.2 Metodologías ágiles

Las metodologías ágiles se pueden definir como las tareas o procedimientos, que están dirigidos a la gestión de proyectos y que incluye la aplicación de métodos de desarrollo, en los que las necesidades y las soluciones puedan evolucionar con el pasar del tiempo, a través del trabajo consensuado de los equipos multidisciplinares. Dichos equipos, se caracterizan por poseer las siguientes cualidades:

² El testing de software, es la realización de pruebas sobre el mismo, con el fin de obtener información acerca de su calidad. <https://openwebinars.net/blog/que-es-el-testing-de-software/> - Consulta realizada el 11/11/2021.

³ En Ingeniería de software, el término Refactorización se usa a menudo para describir la modificación del código fuente sin cambiar su comportamiento, lo que se conoce informalmente por limpiar el código. Es parte del proceso de desarrollo de software. <https://es.wikipedia.org/wiki/Refactorizaci%C3%B3n> - Consulta realizada el 11/11/2021.

⁴ La integración continua es una práctica muy común sobre todo en el ámbito del desarrollo de software ágil. El objetivo de este moderno enfoque, es trabajar a pasos pequeños con el fin de lograr un proceso de desarrollo más efectivo y poder reaccionar con más flexibilidad antes los cambios. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/integracion-continua/> - Consulta realizada el 11/11/2021.

- Evolutivo y flexible.
- Autónomo.
- Planificado.
- Comunicador.

Podemos destacar las siguientes metodologías como las más utilizadas: Scrum, XP y Kanban, según (ALVAREZ, y otros, 2017).

Cabe mencionar, que todas las metodologías ágiles están alineadas con el Manifiesto Ágil (AgileManifesto.org, 2001), el cual se compone por doce principios, agrupados en cuatro valores fundamentales:

A. Individuos e interacciones sobre procesos y herramientas:

En cualquier proyecto ágil, el énfasis debe estar en las personas y en sus interacciones, en vez de los complicados procesos y herramientas.

B. Software funcionando sobre documentación extensiva:

En lugar de la documentación exhaustiva y muy detallada, el enfoque ágil prefiere la entrega incremental de un software funcional, a lo largo del ciclo de vida del producto.

C. Colaboración con el cliente sobre negociación contractual:

El equipo de desarrollo y el cliente, trabajan unidos para evolucionar y desarrollar el producto.

D. Respuesta ante el cambio sobre seguir un plan:

Permite la incorporación de cambios y ciclos de vida rápidos en el desarrollo del producto de manera adaptativa, en lugar de enfatizar el seguimiento de algún plan formado probablemente con información obsoleta o desfasada.

(Yanina Muradas, 2018), resume a las 3 metodologías ágiles más utilizadas a continuación:

Programación Extrema (XP): Conocida por sus siglas XP (*eXtreme Programming*), es una metodología basada en un conjunto de reglas y buenas prácticas para el desarrollo de software, en ambientes muy cambiantes con requisitos imprecisos; por ende, está enfocada en la retroalimentación continua, entre el equipo de desarrollo y el cliente.

Es por ello que, iniciando el proyecto, se deben definir todos los requisitos, para luego invertir el esfuerzo en manejar los cambios que se presenten y así minimizar las posibilidades de error. XP tiene como base la simplicidad y como objetivo, la satisfacción del cliente.

Figura 01: Valores de XP

VALORES DE XP



Fuente: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

Características de XP

En resumen, las principales características de la programación extrema son:

- Desarrollo iterativo e incremental.
- Programación en parejas.
- Pruebas unitarias continuas.
- Corrección periódica de errores.
- Integración del equipo de programación con el cliente.
- Simplicidad, propiedad del código compartida y refactorización del código.

Figura 02: Flujo de XP



Fuente: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

La programación extrema, optimiza los tiempos y se adapta al desarrollo de sistemas grandes y pequeños, sin mayor documentación; el código es claro y simple. Asimismo, complementa los conocimientos entre los miembros del equipo, gracias a la programación en parejas. Sin embargo, una desventaja de esta metodología ágil, es que, luego de cada entrega, el sistema puede ir creciendo, según sean las peticiones del cliente.

Scrum: Esta metodología, es un marco de trabajo de procesos ágiles, que trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto por pares de forma periódica, aplicando las buenas prácticas de trabajo colaborativo (en equipo), facilitando el hallazgo de soluciones óptimas a los problemas que pueden ir surgiendo en el proceso de desarrollo del proyecto.

Con Scrum, se realizan entregas regulares y parciales (*sprint*) del producto final. Todas ellas con una prioridad previamente establecida, que nace según el beneficio que aporten al cliente, minimizando los riesgos que puedan surgir de desarrollos extremadamente largos. Es

por tal motivo, que Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesitan obtener resultados de manera inmediata y donde son fundamentales los siguientes aspectos: la innovación, la productividad, la flexibilidad y la competitividad.

En los Equipos Scrum, se cuenta con roles específicos y cada uno de ellos es imprescindible, para que se lleve a cabo el proceso de forma satisfactoria.

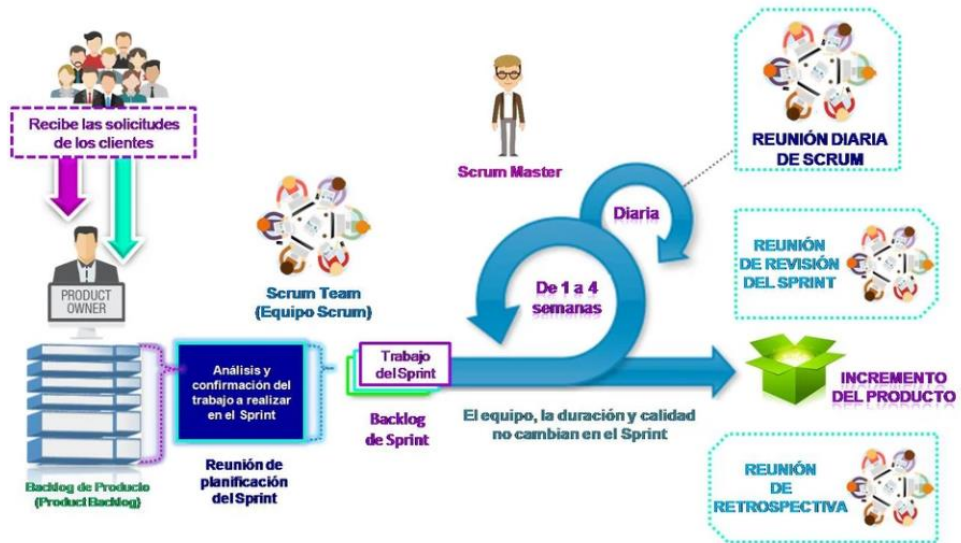
Stakeholder: Es el cliente. Su responsabilidad radica en definir los requerimientos (*Product Backlog*), recibir el producto al final de cada iteración y proporcionar el *feedback* correspondiente.

Product Owner: Es el intermediario de la comunicación entre el cliente (*stakeholder*) y el equipo de desarrollo. Este debe priorizar los requerimientos, según sean las necesidades de la solicitud.

Scrum Master: Actúa como facilitador ante todo el equipo de desarrollo, elimina todos aquellos impedimentos que identifique durante el proceso. Asimismo, se encarga de que el equipo siga los valores y los principios ágiles, las reglas y los procesos de Scrum, incentivando al grupo de trabajo.

Scrum Team (Equipo de desarrollo): Se encarga de desarrollar los casos de uso definidos en el *Product Backlog*. Es un equipo autogestionado, lo que quiere decir que no existe un jefe de equipo, motivo por el cual, todos los miembros se deben de encargar de realizar las estimaciones y, en base a la velocidad obtenida en las iteraciones, irán construyendo el *Sprint Backlog*.

Figura 03: Esquema de SCRUM



Fuente: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

Un punto fundamental en el proceso que conlleva Scrum, son las revisiones (reuniones). Con ellas, se fomenta la comunicación y transparencia del proceso. Las reuniones que aplican, son:

Reunión de planificación: Se debe realizar al inicio de cada *sprint*, esto con el objetivo de planificar la cantidad de trabajo a la que el equipo se va a comprometer a construir, durante el próximo *sprint*.

Reunión diaria: Son reuniones cuyo lapso tiene un máximo 15 minutos. En ellas, se realiza una retroalimentación de qué se hizo el día de ayer, qué se hará hoy y cuáles han sido los problemas que han surgido hasta el momento. El objetivo, es que el equipo establezca un plan para las próximas 24 horas.

Reunión de revisión: Se lleva a cabo al final de cada *sprint*, en ellas se exponen los puntos completados y los que no.

Reunión de retrospectiva: Una vez culminado un *sprint*, se efectúa esta reunión, que tiene como objetivo que el equipo reflexione y saque como resultado, posibles acciones de mejora. A ella, debe asistir todo el Equipo Scrum (Dueño de Producto, Equipo de Desarrollo y Scrum Master). Es una de las reuniones más importantes, ya que es un espacio de reflexión y mejora continua.

Kanban: Proviene de una palabra japonesa, cuyo significado es “Tarjeta Visual”. Es un marco de trabajo que requiere una comunicación en tiempo real, sobre la capacidad del equipo, utilizado para controlar el avance de trabajo en una línea de producción, en la que se clasifican las tareas en subestatus. Esto, con la intención de determinar los niveles de productividad, en cada fase del proyecto.

Para el desarrollo de software, gracias a su sencillez, KANBAN simplifica la planificación y la asignación de responsabilidades. En un tablero, se representan los procesos del flujo de trabajo, como mínimo deben existir tres columnas (Pendiente, En Progreso, Terminado). La cantidad de tarjetas en estatus pendiente, forma parte de lo solicitado por el cliente y aquellas colocadas en progreso, dependerán de la capacidad del equipo de trabajo.

A continuación, se listan las principales ventajas:

- Planificación de tareas.
- Tiempos de ciclos reducidos.
- Rendimiento del equipo de trabajo.
- Métricas visuales.
- Menos cuellos de botella.
- Entrega continua.

Las metodologías ágiles comparten ciertas características, buscan la interacción de los miembros del grupo de trabajo, siempre con la meta de satisfacer los requisitos del cliente. Estas, no se limitan tan sólo a desarrollos de software; con ellas, se pueden gestionar cualquier tipo de proyectos. Es recomendable que las empresas apliquen estos métodos, para eliminar el desperdicio que generan los esfuerzos sin planificación, las reuniones que consumen tiempo y no generan productividad ante alguna iniciativa, entre otros aspectos.

1.2.3 Metodologías basadas en modelos

Esta metodología permite desarrollar un software en un ambiente agradable y de forma fácil, que se basa en modelos. Cumple con los estándares de calidad y se enfoca hacia los procesos del negocio.

Esta metodología está estructurada en 8 fases, las cuales tienen procesos, técnicas, tareas y actividades que conducen a la buena implementación de la metodología. Inicia con la fase de Modelado de Procesos de Negocio (BPM), sigue con las fases de Gestión de Requisitos, Análisis de requisitos, Desarrollo e Implementación, Pruebas, Despliegue, Operación y Mantenimiento, y Gestión de Proyecto. (ROZO NADER, 2014)

- **Modelado de procesos de negocio (BPM):** En esta fase o nivel, se realizan dos procesos: conocer procesos actuales de la organización y crear modelos de procesos.
- **Gestión de requisitos:** En esta fase o nivel, se realiza el proceso de toma de requisitos.
- **Análisis de requisitos:** Esta fase o nivel, consta de dos procesos: Identificar requisitos y modelar requisitos. Las personas que intervienen en esta fase, deberían ser: administradores del proyecto, aseguradores de calidad, analistas, diseñadores y documentadores.
- **Desarrollo e Implementación:** Esta fase o nivel consta del proceso: Construcción del sistema de información. Las personas que intervienen en esta fase, deberían ser: administrador del proyecto, aseguradores de calidad, diseñadores, programadores, administrador de la configuración y documentadores.
- **Pruebas:** Esta fase se realiza en paralelo con la construcción del sistema de información, debido a que la terminación del código por parte de cada grupo de trabajo, debe pasar por pruebas, antes de ser fusionado. Esta fase o nivel consta de dos procesos: ejecución de pruebas individuales y ejecución de pruebas del sistema.
- **Despliegue:** Esta fase o nivel, consta del proceso de entrega del sistema. Las personas que intervienen en esta fase, deberían ser:

administrador del proyecto, cliente, programadores y documentadores.

- **Operación y Mantenimiento:** Esta fase consta del proceso del seguimiento y control, se destacan los modelos que contienen las mejores prácticas, que ayudan a las organizaciones a mejorar sus procesos, tal como CMMI⁵. Las personas que intervienen en esta fase, deberían ser: administrador del proyecto, aseguradores de calidad, administrador de la configuración, ingeniero de validación y verificación cliente, *testers* y documentadores.
- **Gestión de Proyectos:** Esta fase se compone de cuatro procesos: evaluación de funcionalidad, ejecución de pruebas individuales, ejecución de pruebas del sistema y gestión de proveedores. Las personas que intervienen en esta fase, deberían ser: administrador del proyecto, aseguradores de calidad, cliente, analistas, diseñadores ingenieros de mantenimiento, programadores, administrador de la configuración, ingeniero de validación y verificación y documentadores.

1.2.4. Calidad de Software

El término calidad de software, se refiere al grado de desempeño de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida. Dichas características, de cierta manera, garantizan que el cliente cuente con un sistema confiable, lo cual aumenta su satisfacción frente a la funcionalidad y eficiencia del sistema construido. (Callejas Cuervo, y otros, 2017).

El concepto de calidad de software, según (PRESSMAN, 2010), se asocia a la "concordancia con los requisitos funcionales y de rendimiento, explícitamente establecidos con los estándares de desarrollo plenamente documentados y con las características

⁵ Capability Maturity Model Integration o Integración de Modelos de Madurez de Capacidades (CMMI), es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Desarrollado en la Universidad Carnegie Mellon. https://es.wikipedia.org/wiki/Capability_Maturity_Model_Integration - Consulta realizada el 11/11/2021.

implícitas, que se espera de todo software desarrollado profesionalmente", con base en los requisitos funcionales y no funcionales identificados en la etapa de análisis del sistema, insumo principal para implementar dichos requisitos con los atributos mínimos de calidad, fomentando la aplicación de procesos estandarizados y criterios necesarios en cada una de sus etapas. Así, se fomenta que el avance en el ciclo de vida del software, minimice el riesgo de fracaso del proyecto.

Por su parte, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, 1995), define calidad de software como "el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario", denotando que el énfasis radica en los requisitos específicos del sistema y en la búsqueda de la satisfacción del cliente.

Para garantizar la calidad de software, es importante implementar algún modelo o estándar de calidad, que permita la gestión de atributos en el proceso de construcción de software, teniendo en cuenta que la concordancia de los requisitos y su construcción son la base de las medidas de calidad establecidas.

Aunque modelo y metodología distan en su definición, se rescata la cita dada por (MOSZKOWITZ, 2010), en la que presenta una metodología que permite, a cualquier organización, realizar una autoevaluación o autodiagnóstico, por medio de una revisión sistemática de sus estrategias y prácticas de gestión.

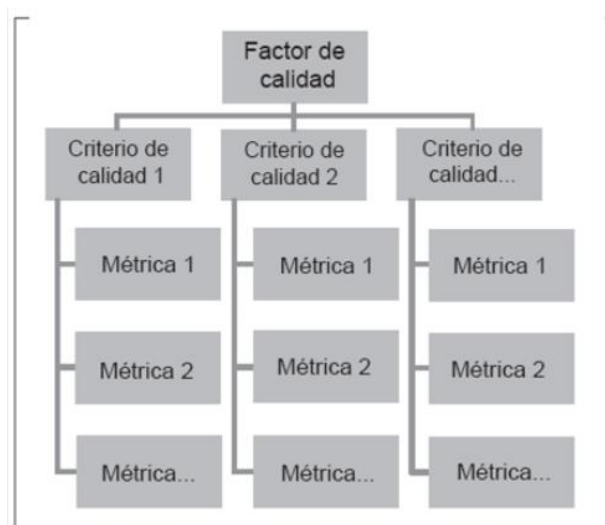
En el caso de la calidad de software, el modelo debe ir enfocado a hacer seguimiento y evaluación a cada etapa de construcción del producto software. Por otro lado, menciona (SCALONE, 2006) que los modelos de calidad, son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad.

Esta definición, enfocada a la calidad del software, identifica que la organización debe contar con un proceso que, como soporte al mismo, lleve una documentación y se valga de distintas prácticas definidas en el modelo, dando apoyo a la organización para tener una mejora continua y ser más competentes, para así poder medir la calidad y brindar productos o servicios de alto nivel.

En el ámbito de la construcción de software, el modelo de calidad debe permitir evaluar el sistema, bien sea cualitativa o cuantitativamente y, de acuerdo con esta evaluación, la organización podrá proponer e implementar estrategias que permitan la mejora del proceso, dentro de las etapas de análisis, diseño, desarrollo y pruebas del software.

Los modelos de calidad de software, generalmente están estructurados como se muestra en la Figura 04 (BAUTISTA, 2012), donde se pueden tener diversos factores de calidad que, a su vez, se componen de criterios que son evaluados por métricas, con el propósito de abordar la evaluación desde lo general a lo particular y permitir la reducción de la subjetividad en la asignación de un valor, ya sea cuantitativo o cualitativo.

Figura 04: Estructura de la calidad de software.



Fuente: Los autores (SCALONE, 2006), (BAUTISTA, 2012).

Asimismo, los modelos de calidad de software, se clasifican de acuerdo con el enfoque de evaluación, ya sea a nivel de proceso, producto o calidad en uso.

- **Calidad a nivel de proceso:** La calidad de un sistema software, debe ser programada desde el inicio del proyecto y, posteriormente, en cada etapa del proceso de desarrollo, se debe llevar a cabo el control y seguimiento de los aspectos de calidad, para minimizar los riesgos y ofrecer soporte continuo. Se garantiza así, un óptimo nivel de cumplimiento de los factores de calidad, teniendo en cuenta que, si en alguna de las etapas se deja de lado la verificación de los factores y criterios, es posible que se presente deficiencia en alguno de éstos y disminuya el nivel de calidad no solo del proceso, sino también del producto en desarrollo.
- **Calidad a nivel de producto:** La principal finalidad del modelo de calidad de producto, es especificar y evaluar el cumplimiento de criterios del producto, para lo cual se aplican medidas internas y/o medidas externas (BEVAN, 2010). Por esta razón, algunas normas y estándares, han definido la calidad a nivel de producto en tres tipos: interna, externa y en uso (RODRIGUEZ, 2016). Este enfoque, está orientado a verificar el cumplimiento de las características, que permitan alcanzar la satisfacción del cliente, en cuanto a los requisitos definidos en las etapas iniciales del proceso de desarrollo.
- **Calidad en uso:** Es importante resaltar que, aunque en diferentes escenarios, se utilizan los términos usabilidad y calidad en uso, con el mismo propósito y de forma intercambiable. Tienen significados distintos, principalmente, porque el concepto de calidad en uso es más amplio y abarca más elementos que la usabilidad (COVELLA, 2005), y, esta última, es una de las características de calidad de un producto software. La calidad en uso, se define como el "conjunto de atributos relacionados con la aceptación por parte del usuario final y seguridad" y está basada en la eficacia, productividad, seguridad y satisfacción, según ISO/IEC 9126.

Modelos a nivel de proceso

- **ITIL:** Desarrollado en el Reino Unido con el fin de fortalecer la gestión gubernamental, a partir de cinco elementos fundamentales: la perspectiva del negocio, entrega del servicio, soporte del servicio, manejo de la infraestructura y manejo de aplicaciones. También, con el propósito de ofrecer una estructura integral, para prestar a la organización un servicio completo, cubriendo necesidades de apoyo de instalación, adecuación de redes, comunicaciones, hardware, servidores, sistema operativo y software necesarios.
- **ISO/IEC 15504:** Permite adaptar la evaluación para procesos en pequeñas y medianas empresas (pymes) y grupos de desarrollo pequeños, mediante la estructuración en seis niveles de madurez: Nivel 0 - Organización inmadura, Nivel 1 - Organización básica, Nivel 2 - Organización gestionada, Nivel 3 - Organización establecida, Nivel 4 - Organización predecible y Nivel 5 - Organización optimizando. Su objetivo es llegar a que la organización logre ser madura, lo cual conlleva a que la organización tenga procesos definidos, responsabilidades definidas, predicción de resultados, productos entregados con calidad, que las entregas se den en los tiempos pactados, incrementar la productividad, clientes satisfechos y empleados felices.
- **Bootstrap:** Es una metodología de evaluación, que permite la mejora de procesos a partir de seis actividades básicas: Examinar la necesidad, iniciar proceso de mejora, preparación y dirección de la evaluación, análisis de resultados, implantación y finalización de mejoras.
- **Dromey:** Es un modelo adaptable, para evaluar varias etapas del proceso de desarrollo, como levantamiento de requisitos, diseño e implementación. Se estructura con características y subcaracterísticas de calidad. Propone tres modelos distintos para cada etapa de construcción del producto: modelo de requerimientos, modelo de diseño y modelo de calidad de la implementación, a partir de la evaluación establecida en cinco etapas, para características

como: eficiencia, confiabilidad, mantenibilidad, portabilidad, facilidad de uso y funcionalidad.

- **Personal Software Process (PSP):** Este modelo está enfocado al desarrollo profesional del ingeniero, fomentando una adecuada administración de calidad de los proyectos de desarrollo, reducción de defectos del producto, estimación y planeación del trabajo.
- **Team Software Process (TSP):** TSP es la fase posterior de PSP. Está diseñado para el trabajo de equipos de desarrollo de software autodirigidos, que se orienta al desarrollo de productos con el mínimo de defectos en tiempo y costos estimados. Cuenta con planes detallados y procesos, como revisiones personales, inspecciones e índices de desempeño de calidad y el fomento de la integración del equipo.
- **IEEE / EIA 12207:** Este estándar, establece un marco de trabajo común, para el ciclo de vida del desarrollo de software, a partir del planteamiento de procesos, actividades y tareas que pueden ser aplicadas durante la adquisición, suministro, desarrollo, operación, mantenimiento y/o despliegue de un producto software.
- **Cobit 4.0:** Se caracteriza por ser orientado a negocios y procesos. Además de ser basado en controles, trabaja con siete criterios de información, que son definidos como requerimientos de control del negocio: efectividad, eficiencia, confidencialidad, integridad, disponibilidad, cumplimiento y confiabilidad.
- **ISO 90003:** Conjunto de estándares utilizados para el desarrollo, suministro y soporte del software, cuyo propósito es ofrecer una guía de aplicación de la norma 9001, que pretende ser utilizada para demostrar o soportar, que la entidad está en capacidad de desarrollar software con criterios de calidad.
- **CMMI (Capability Maturity Model Integration):** Es de los modelos más utilizados en las empresas de construcción de software, con el propósito de verificar el cumplimiento de estándares de calidad, a partir de la medición con niveles de madurez. Este modelo se representa de dos maneras: escalonada y continua, donde el modelo

escalonado está dirigido al software y permite clasificar las organizaciones en cinco tipos de nivel establecidos: Inicial, gestionado, definido, gestionado cuantitativamente y en optimización. Por su parte, el modelo continuo se enfoca en el análisis de la capacidad de cada proceso inmerso en las áreas de la ingeniería de sistemas y lo clasifica en uno de los siguientes seis niveles: Incompleto (0), ejecutado (1), gestionado (2), definido (3), cuantitativamente gestionado (4) y en optimización (5).

- **ISO/IEC 20000:** El objetivo principal de esta norma, es el de avalar que la prestación de servicios gestionados de TI de una empresa, cuenta con la calidad necesaria para brindar dichos servicios a los clientes. Se subdivide en dos partes: "Especificaciones", publicada como ISO 200001:2005 y "Código de buenas prácticas", publicada como ISO 20000-2:2005.

Modelos a nivel de producto

- **McCall:** Es uno de los modelos pioneros en la evaluación de la calidad de software. Tiene tres etapas definidas: factores, criterios y métricas. Los once criterios base, son: Exactitud, confiabilidad, eficiencia, integridad, usabilidad, mantenibilidad, testeabilidad, flexibilidad, portabilidad, reusabilidad e interoperabilidad.
- **GQM o Goal Question Metric:** Se enfoca a proporcionar una forma que permita definir métricas, para medir el avance como los resultados de algún proyecto, a partir de la aplicación de unas preguntas relacionadas con el proyecto, que permitan alcanzar unas metas previamente planteadas. El modelo trabaja sobre metas, preguntas y métricas.
- **Boehm:** Es un modelo incremental, dividido en regiones de tareas y estas, a su vez, en conjuntos de tareas, las cuales se ajustan a la cantidad de iteraciones que el equipo defina. Cada iteración, se divide en cuatro sectores: planeación, análisis de riesgo, ingeniería y evaluación.

- **FURPS:** Es un modelo desarrollado por Hewlett-Packard, cuyo nombre proviene de los criterios que evalúa: Funcionalidad, usabilidad, confiabilidad (*reliability*), desempeño (*performance*) y soportabilidad.
- **GILB:** Es un modelo de calidad, que orienta la evaluación de software a partir de los atributos: Capacidad de trabajo, adaptabilidad, disponibilidad y utilizabilidad. Estos, se dividen en subatributos, de tal manera, que sirva de apoyo a la gestión de proyectos y proporcione una guía para solucionar problemas y detectar riesgos.
- **ISO 9126:** Estándar basado en el modelo de McCall, dirigido a desarrolladores, aseguradores de calidad, evaluadores, analistas y cualquier otro involucrado en el proceso de construcción de software. Está dividido en cuatro partes: modelo de calidad, métricas externas, métricas internas y calidad de métricas en uso. También, en elementos en torno a seis características (funcionalidad, habilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y sub características asociadas.
- **SQAE o Software Quality Assessment Exercise:** Este modelo, basado en Boehm, McCall, Dromey e ISO 9126, está orientado, principalmente, a realizar evaluación por terceros, que no están directamente involucrados con el desarrollo, siguiendo tres capas: área, factor y atributo de calidad, que permiten orientar la evaluación jerárquicamente.
- **WebQEM:** Es una metodología de evaluación de calidad de sitios Web (*Web-site Quality Evaluation method*), diseñada para la evaluación siguiendo seis fases: planificación y programación de la evaluación de calidad, definición y especificación de requerimientos de calidad, definición e implementación de la evaluación elemental, definición e implementación de la evaluación global, análisis de resultados, conclusión y documentación, y validación de métricas.
- **ISO 25000:** También llamada como SQuaRE, cuyo propósito es guiar el desarrollo con los requisitos y la evaluación de atributos de calidad; principalmente: la adecuación funcional, eficiencia de desempeño,

compatibilidad, capacidad de uso, habilidad, seguridad, mantenibilidad y portabilidad.

1.3. DEFINICIÓN DE TÉRMINOS BÁSICOS

- **Ingeniería de software:** La Ingeniería de Software es una de las ramas de las ciencias de la computación, que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería. Brinda soporte operacional y de mantenimiento. Integra ciencias de la computación, ciencias aplicadas y las ciencias básicas, en las cuales se encuentra apoyada la ingeniería. (Wikipedia, 2020).
- **Metodología:** Se define como el enfoque de un problema de manera total, organizada, sistemática y disciplinada. Esta definición, muestra una distinción entre “metodología” y “técnica”. La técnica se considera como un componente de la metodología, como el medio o procedimiento que se usa, para realizar la metodología misma. (ROZO NADER, 2014).
- **Rational Unified Process (RUP):** El Proceso Unificado es un proceso de software genérico, que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos. Provee un enfoque disciplinado en la asignación de tareas y responsabilidades, dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad, que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. (TECHLIB, 2018).
- **Gestión de procesos de negocio (BPM):** BPM es un enfoque de manejo adaptable, desarrollado con el fin de sistematizar y facilitar los procesos individuales de negocio complejos, dentro y fuera de las empresas. BPM tiene como objetivo, traer a colación la información relevante sobre cómo los procesos se ejecutan de manera que se puedan hacer mejoras y para que los procesos se puedan manejar,

permitiendo una mejor toma de decisiones y visión de negocios como un todo. (HEFLO, 2020).

- **Patrones de diseño:** Los patrones de diseño, son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. No se puede elegir un patrón y copiarlo en el programa, como si se tratara de funciones o bibliotecas ya preparadas. El patrón no es una porción específica de código, sino un concepto general para resolver un problema particular. A menudo, los patrones se confunden con algoritmos, porque ambos conceptos describen soluciones típicas a problemas conocidos. Mientras que un algoritmo siempre define un grupo claro de acciones para lograr un objetivo, un patrón es una descripción de más alto nivel de una solución. El código del mismo patrón aplicado a dos programas distintos, puede ser diferente. (GURU, 2014)
- **Principios SOLID:** Hablando de diseño y desarrollo de aplicaciones, Principios SOLID son unas palabras que se conocen como uno de los fundamentos de la arquitectura y desarrollo de software. Los 5 principios SOLID de diseño de aplicaciones de software, son:
 - S – Single Responsibility Principle (SRP)
 - O – Open/Closed Principle (OCP)
 - L – Liskov Substitution Principle (LSP)
 - I – Interface Segregation Principle (ISP)
 - D – Dependency Inversion Principle (DIP)

Entre los objetivos de tener en cuenta estos 5 principios a la hora de escribir código, encontramos:

- Crear un software eficaz, que cumpla con su cometido y que sea robusto y estable.
- Escribir un código limpio y flexible ante los cambios, que se pueda modificar fácilmente según necesidad, que sea reutilizable y mantenible.
- Permitir escalabilidad, que acepte ser ampliado con nuevas funcionalidades de manera ágil.

En definitiva, desarrollar un software de calidad. (PROFILE, 2018)

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. DESCRIPCIÓN DEL PROBLEMA

Los cambios en el campo de la tecnología y el impacto que tiene en los negocios, han crecido exponencialmente en los últimos años. Nunca, la tasa de cambio ha sido tan rápida como ahora. Para no quedarse atrás, las empresas necesitan construir nuevas capacidades y aprovechar los cambios del entorno.

Muchas de las empresas, buscan cubrir sus necesidades, creando aplicaciones propias o sistemas pequeños de manera artesanal, sin seguir una metodología en particular o mezclando varias de ellas. Es decir, si es que el grupo de desarrolladores llegase a usar una de las metodologías, esta se termina adaptando a la empresa. Sin embargo, esto debería ser al revés: las empresas deberían adaptarse a las metodologías de desarrollo de software.

El común denominador de muchas empresas que desarrollan software, es que este desarrollo se hace de manera artesanal, a pesar de la existencia de metodologías estructuradas y probadas para el desarrollo de software. Esto, debido a diversos factores tales como la falta de conocimiento, la poca estandarización de las políticas de la empresa sobre el desarrollo de aplicaciones, la moda que impone el uso de una nueva metodología de desarrollo de software o, también, debido a que las empresas podrían considerar que, seguir una metodología formal, significa una pérdida de recursos, tanto de tiempo, como de personas en lo referente a la documentación y capacitación.

A pesar de existir metodologías estructuradas, con plantillas, indicadores y métricas, las empresas terminan por alterarlas, mezclando varias en una sola o simplemente desarrollándolas de manera empírica.

2.2. FORMULACIÓN DEL PROBLEMA

2.2.1. Problema general

¿Cómo influye el uso de metodologías de desarrollo de software, en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021?

2.2.2 Problemas Específicos

- ¿De qué manera el uso de metodologías de desarrollo de software, promueve el trabajo efectivo en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021?
- ¿De qué manera el uso de metodologías de desarrollo de software, permite disminuir los costos de mantenimiento de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021?
- ¿De qué manera el uso de metodologías de desarrollo de software, permite una rápida respuesta a los cambios en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021?

2.3. OBJETIVOS

2.3.1. Objetivo general

Describir la manera en cómo influye el uso de metodologías de desarrollo de software, en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.

2.3.2. Objetivos específicos

- Describir la manera en que el uso de metodologías de desarrollo de software, promueve el trabajo efectivo en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.
- Describir la manera en que el uso de metodologías de desarrollo de software, permite disminuir los costos de mantenimiento de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.
- Describir la manera en que el uso de metodologías de desarrollo de software, permite una rápida respuesta a los cambios en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.

2.4 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Dado el carácter multidisciplinario de los sistemas de información, se adquieren conocimientos y se desarrollan habilidades de muy diversa índole. A las capacidades propias de un ingeniero de sistemas, centradas en el desarrollo e integración de aplicaciones informáticas, se añaden las específicas, relacionadas con los aspectos organizacionales, empresariales, de interacción y de comunicación, propios de los sistemas de información.

El objetivo principal de la ingeniería del software, es obtener productos de alta calidad; para ello, es necesario evaluar los productos software que se produzcan en la empresa, para mejorar la calidad del servicio ofertado al cliente. La calidad requiere de control y, a su vez, se hace necesario un método aceptado que permita diagnosticar, en forma ordenada, aquello que deseamos resolver o mejorar.

Las Metodologías de Desarrollo de Software, surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto software. Dichas metodologías, pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software.

La presente investigación es importante, porque se enfoca en el proceso de desarrollo de software, para determinar alguna influencia del uso de metodologías de desarrollo de software en la calidad del software producido. Esto, debido al último Reporte CHAOS⁶ 2020 del Standish Group International, donde se indica que tan sólo el 31% de los proyectos evaluados fueron exitosos; es decir, cumplieron los objetivos del proyecto (Tiempo, Alcance, Costo, y Calidad).

⁶ El reporte del Caos (CHAOS Report), viene siendo publicado por Standish group desde 1994 dando una visión sobre el fracaso o éxito de los proyectos.

<https://hennyporntman.files.wordpress.com/2021/01/project-success-grc-standish-group-chaos-report-2020.pdf>

En los últimos años, se han realizado diversos esfuerzos por mejorar el proceso de desarrollo de software, se integraron practicas universales de gestión de proyectos de TI, se refinaron los ciclos de vida de software, la asociación de estandarización “IEEE” liberó el libro de conocimiento de la ingeniera de software (SWEBOOK). Adicionalmente, se crearon diversos marcos de trabajo CMMI, ISO 12207, etc., metodologías de desarrollo de software, pero, desafortunadamente, estos esfuerzos no han tenido el impacto deseado.

2.5 HIPÓTESIS

2.5.1 Hipótesis General

El uso de metodologías de desarrollo de software, influye en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.

2.5.2. Hipótesis específicas

- El uso de metodologías de desarrollo de software, promueve el trabajo efectivo en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.
- El uso de metodologías de desarrollo de software, permite disminuir los costos de mantenimiento de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.
- El uso de metodologías de desarrollo de software, permite una rápida respuesta a los cambios en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021.

2.6. VARIABLES

2.6.1. Identificación de las variables

- ✓ **Independiente (X):** Metodologías de desarrollo de software.
- ✓ **Dependiente (Y):** Calidad del software.

2.6.2. Definición de las variables

La variable independiente (X): Metodologías de desarrollo de software, se define como el modo sistemático de realizar, gestionar y administrar un proyecto, para llevarlo a cabo con altas probabilidades de éxito.

La variable dependiente (Y): Calidad del software, se define como el conjunto de cualidades, que caracterizan al software y que determinan su utilidad y existencia.

2.6.3. Operacionalización de las variables

Tabla 01: Operacionalización de variables

Variables	Dimensión	Indicadores	Índices	Escala de medición
Independiente (X): Metodologías de desarrollo de software	Efectividad	Nivel de comunicación entre los interesados del proyecto	Muy alto, Alto, Medio, Bajo, Muy bajo.	Ordinal
		Nivel de capacidad de auto gestión del equipo de trabajo		
	Costo	Nivel de costo de conformidad		
		Nivel de costo de no conformidad		
	Flexibilidad	Nivel de respuesta a cambios en los procesos de la organización.		
		Nivel de adaptabilidad del software a los requisitos cambiantes del proyecto.		
Dependiente (Y): Calidad del software	Usabilidad	Nivel de facilidad en el aprendizaje del uso del software	Muy alto, Alto, Medio, Bajo, Muy bajo.	Ordinal
		Nivel de facilidad en el uso operativo del software		
	Fiabilidad	Nivel de tolerancia a fallos del software		
		Nivel de exactitud en el procesamiento de la información		
	Mantenibilidad	Nivel de facilidad para localizar un fallo operativo del software		
		Nivel de facilidad para arreglar un fallo operativo del software		

Fuente: Elaboración propia.

CAPÍTULO III METODOLOGÍA

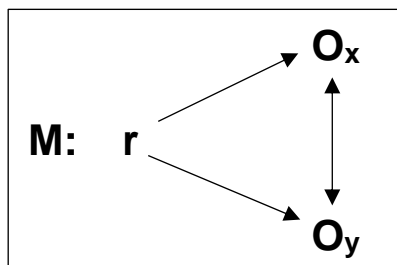
3.1 TIPO Y DISEÑO DE INVESTIGACIÓN

3.1.1. Tipo de investigación

El presente proyecto, utilizó el tipo de investigación descriptiva, puesto que describió la realidad particular de la situación que se abordó y analizó, tal como se ha descrito en la problemática.

3.1.2. Diseño de investigación

La investigación tuvo un diseño no experimental transversal de tipo correlacional, donde se observaron las variables en un momento dado de la investigación, para determinar la posible asociación entre las mismas.



Donde:

M: Muestra

O_x: Observación a la variable independiente: Metodologías de desarrollo de software

O_y: Observación a la variable dependiente: Calidad de software

r: Posible asociación entre las variables.

3.2. POBLACIÓN Y MUESTRA

3.2.1. Población

La población tomada en cuenta para la investigación corresponde a 11 individuos, que representan al 100% de las personas que laboran en la empresa 'Software y Sistemas del Perú S.A.C'. – Iquitos.

3.2.2. Muestra

La muestra fue de tipo no probabilística por conveniencia y estuvo conformada por 11 individuos, basándose en la fórmula que se describe a continuación:

$$n = \frac{z^2(p \cdot q)}{e^2 + \frac{z^2(p \cdot q)}{N}}$$

Donde:

Tamaño de la muestra (**n**): 11

Nivel de confianza deseado (**z**): 95%

Proporción de la población con la característica deseada (**p**): 11

Proporción de la población sin la característica deseada (**q**): 11

Margen de error (**e**): 5%

Población (**N**): 11

3.2.3. Participantes

Tabla 02: Trabajadores encuestados

Ítem	Nombre	Cargo	Profesión
1	Edgard Ocampo Fasabi	Programador Backend	Ing. De Sistemas
2	Erick José Jara Palacios	Analista de Sistemas	Ing. De Sistemas
3	Mathew Guevara Escobedo	Analista en Base de datos	Ing. De Sistemas
4	Gerson Ríos Pinedo	Programador Frontend	Bachiller en Ing. De Sistemas
5	Saulo Patrick Linares Chumbe	Programador Backend	Bachiller en Ing. De Sistemas
6	Pedro Jarry Rodríguez Quiroz	Jefe de Soporte Técnico	Técnico en Computación e informática
7	Ronald Alberto Grónerth Álvarez	Programador Backend y Frontend	Técnico en Computación e informática
8	Juana Dora Pupuche Isuiza	Soporte Técnico	Técnico en Computación e informática
9	Geny Silvina Vásquez Ahuanari	Soporte Técnico	Técnico en Computación e informática
10	Kasandra Cachique Sinti	Analista Técnico	Técnico en Computación e informática
11	Katherine Núñez Tuanama	Jefe de Proyecto	Contador, Técnico en computación

Fuente: Elaboración propia.

3.3. TÉCNICAS, INSTRUMENTOS Y PROCEDIMIENTOS DE RECOLECCIÓN DE DATOS

3.3.1. Técnica de recolección de datos

- **Encuesta:** Es una técnica que se desarrolla aplicando un cuestionario, a una determinada muestra de personas. Esto suministra información sobre las opiniones, actitudes y comportamiento de las personas participantes. Esta técnica, se aplica ante la necesidad de probar una o varias hipótesis o descubrir una solución a un problema en particular.

3.3.2. Instrumento de recolección de datos

Para esta investigación, el instrumento utilizado fue el cuestionario y estuvo dirigido a todos los sujetos de la muestra. Las preguntas contenidas, se elaboraron de acuerdo a las dimensiones vinculadas a las variables, guardando coherencia con el problema, hipótesis y objetivos.

Para el cuestionario de esta investigación, se utilizó la escala de Likert, la cual es una escala de calificación que se utiliza para cuestionar a una persona sobre su nivel de acuerdo o desacuerdo con una declaración. Es ideal para medir reacciones, actitudes, comportamientos y opiniones de una persona. (Hernandez Sampieri, y otros, 1997).

Para cada pregunta del cuestionario, se dieron 5 alternativas de respuesta, que fueron: Muy alto, Alto, Medio, Bajo, Muy bajo. Los puntajes asignados para cada alternativa, fueron:

- Muy alto = 5
- Alto = 4
- Medio = 3
- Bajo = 2
- Muy bajo = 1

3.3.3. Procedimiento de recolección de datos

El procedimiento para la recolección de datos, se desarrolló de la siguiente manera:

- Elaborar el instrumento de recolección de datos.
- Realizar pruebas de validez y confiabilidad al instrumento de recolección de datos.
- Aplicar la encuesta a la muestra seleccionada.
- Realizar el procesamiento y análisis de los datos obtenidos.
- Interpretar los datos.
- Elaborar la discusión e informe final.
- Sustentar los resultados del informe final.

3.4. PROCESAMIENTO Y ANÁLISIS DE DATOS

3.4.1. Procesamiento de los datos

El procesamiento de los datos, se realizó a través de la estadística descriptiva, aplicando tablas, cuadros, gráficos y figuras. Los datos se analizaron, utilizando un software estadístico especializado.

3.4.2. Análisis de los datos

Para la realización de la inferencia estadística, se aplicó una prueba de normalidad. De acuerdo a los resultados, se optó por la utilización de los estadísticos de prueba T-Student o Chi Cuadrada, según el caso.

CAPÍTULO IV RESULTADOS

4.1 RESULTADOS

4.1.1 Estadística Descriptiva

Tabla 03: Estadística descriptiva para el índice Muy Alto (5)

Muy Alto (5)	
Media	6.583333333
Error típico	0.416666667
Mediana	7
Moda	8
Desviación estándar	2.041241452
Varianza de la muestra	4.166666667
Curtosis	1.018543196
Coefficiente de asimetría	-1.117974228
Rango	8
Mínimo	1
Máximo	9
Suma	158
Cuenta	24
Nivel de confianza (95.0%)	0.861940671

Fuente: Elaboración propia.

Tabla 04: Estadística descriptiva para el índice Alto (4)

Alto (4)	
Media	7.083333333
Error típico	0.334689271
Mediana	7.5
Moda	8
Desviación estándar	1.639635873
Varianza de la muestra	2.688405797
Curtosis	3.033033139
Coefficiente de asimetría	-1.565303525
Rango	7
Mínimo	2
Máximo	9
Suma	170
Cuenta	24
Nivel de confianza (95.0%)	0.692357508

Fuente: Elaboración propia.

Tabla 05: Estadística descriptiva para el índice Medio (3)

Medio (3)	
Media	6.125
Error típico	0.325890641
Mediana	6
Moda	6
Desviación estándar	1.596531567
Varianza de la muestra	2.548913043
Curtosis	0.554238039
Coefficiente de asimetría	-0.501547601
Rango	7
Mínimo	2
Máximo	9
Suma	147
Cuenta	24
Nivel de confianza (95.0%)	0.674156155

Fuente: Elaboración propia.

Tabla 06: Estadística descriptiva para el índice Bajo (2)

Bajo (2)	
Media	2.5
Error típico	0.306924907
Mediana	2
Moda	1
Desviación estándar	1.503618823
Varianza de la muestra	2.260869565
Curtosis	-0.292294244
Coefficiente de asimetría	0.795284135
Rango	5
Mínimo	1
Máximo	6
Suma	60
Cuenta	24
Nivel de confianza (95.0%)	0.634922545

Fuente: Elaboración propia.

Tabla 07: Estadística descriptiva para el índice Muy Bajo (1)

Muy Bajo (1)	
Media	1.708333333
Error típico	0.265662739
Mediana	1
Moda	1
Desviación estándar	1.301476308
Varianza de la muestra	1.69384058
Curtosis	12.17629121
Coefficiente de asimetría	3.177499226
Rango	6
Mínimo	1
Máximo	7
Suma	41
Cuenta	24
Nivel de confianza (95.0%)	0.549565247

Fuente: Elaboración propia

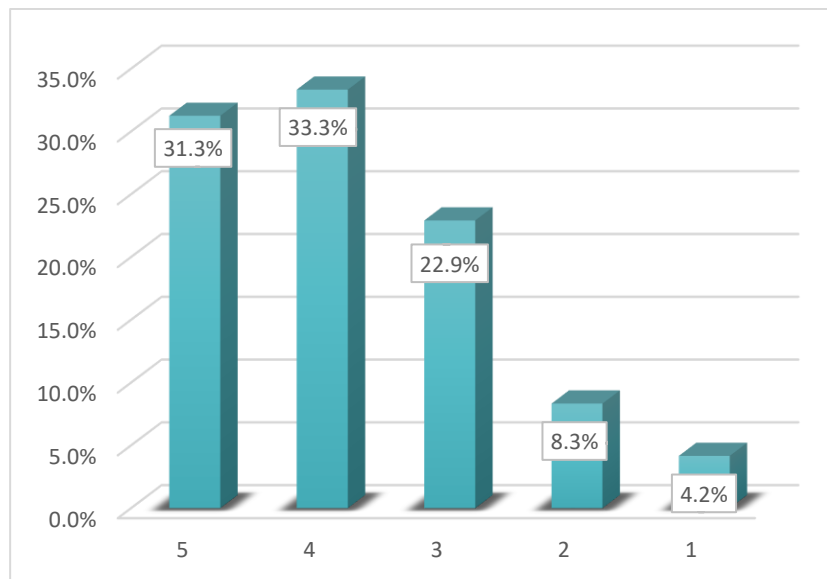
4.1.2. Análisis de los resultados de la variable independiente – Metodologías de desarrollo de software

Tabla 08: Efectividad - Nivel de comunicación entre los interesados del proyecto

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	7.5	31.3%	31.3%
Alto	8	33.3%	64.6%
Medio	5.5	22.9%	87.5%
Bajo	2	8.3%	95.8%
Muy Bajo	1	4.2%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 01: Efectividad - Nivel de comunicación entre los interesados del proyecto



Fuente: Elaboración propia

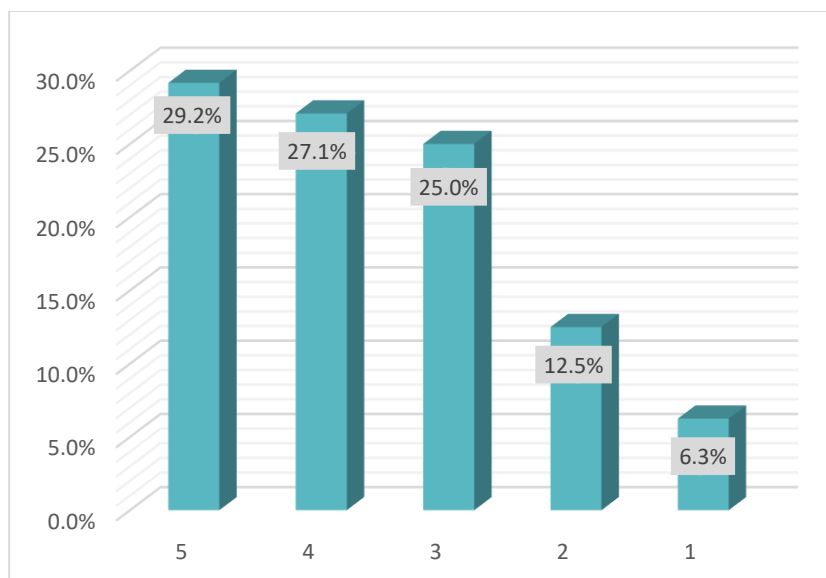
Según los resultados de la Tabla 08 y Gráfico 01, observamos un alto nivel de comunicación entre los interesados del proyecto, en la empresa analizada, representando un 64.6% de los encuestados (sumando Índice Muy alto y Alto). Un 22.9% de los encuestados, indican un índice Medio, mientras que un 8.3%, se encuentran en índice Bajo. Finalmente, un 4.2% de los encuestados, manifestaron un índice Muy Bajo.

Tabla 09: Efectividad - Nivel de capacidad de autogestión del equipo de trabajo

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	7	29.2%	29.2%
Alto	6.5	27.1%	56.3%
Medio	6	25%	81.3%
Bajo	3	12.5%	93.8%
Muy Bajo	1.5	6.3%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 02: Efectividad - Nivel de capacidad de autogestión del equipo de trabajo



Fuente: Elaboración propia

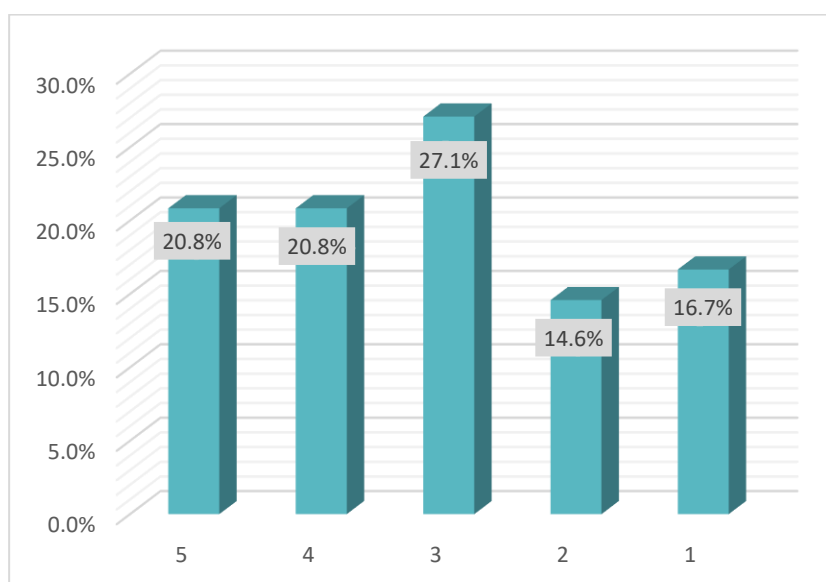
En cuanto al nivel de capacidad de autogestión del equipo de trabajo, expresados en la Tabla 09 y Gráfico 02, se registra un 29.2% de respuestas en el índice Muy Alto, un 27.1% en el índice Alto y un 25% en el índice Medio. Mientras que los índices Bajo y Muy Bajo, registran un 12.5% y 6.3%, respectivamente.

Tabla 10: Costo - Nivel de costo de conformidad

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	5	20.8%	20.8%
Alto	5	20.8%	41.7%
Medio	6.5	27.1%	68.8%
Bajo	3.5	14.6%	83.3%
Muy Bajo	4	16.7%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 03: Costo - Nivel de costo de conformidad



Fuente: Elaboración propia

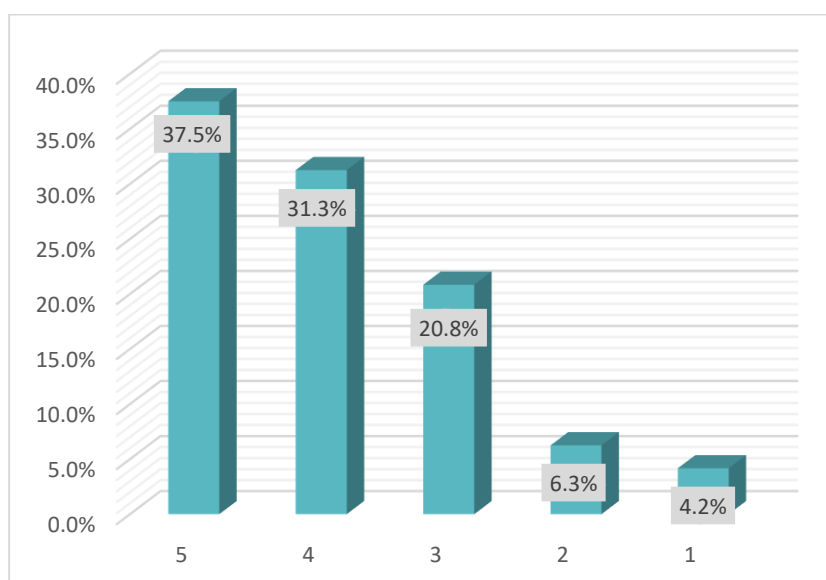
En cuanto al análisis del nivel de costo de conformidad, la Tabla 10 y Gráfico 03, nos muestra que el índice Medio, obtuvo el mayor resultado con un 27.1% de los encuestados. Los índices Muy Alto y Alto, obtuvieron ambos un 20.8%. En similar proporción, un 14.6% y 16.7% de los encuestados, indicaron un índice Bajo y Muy Bajo, respectivamente.

Tabla 11: Costo - Nivel de costo de no conformidad

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	5	37.5%	37.5%
Alto	5	31.3%	68.8%
Medio	6.5	20.8%	89.6%
Bajo	3.5	6.3%	95.8%
Muy Bajo	4	4.2%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 04: Costo - Nivel de costo de no conformidad



Fuente: Elaboración propia.

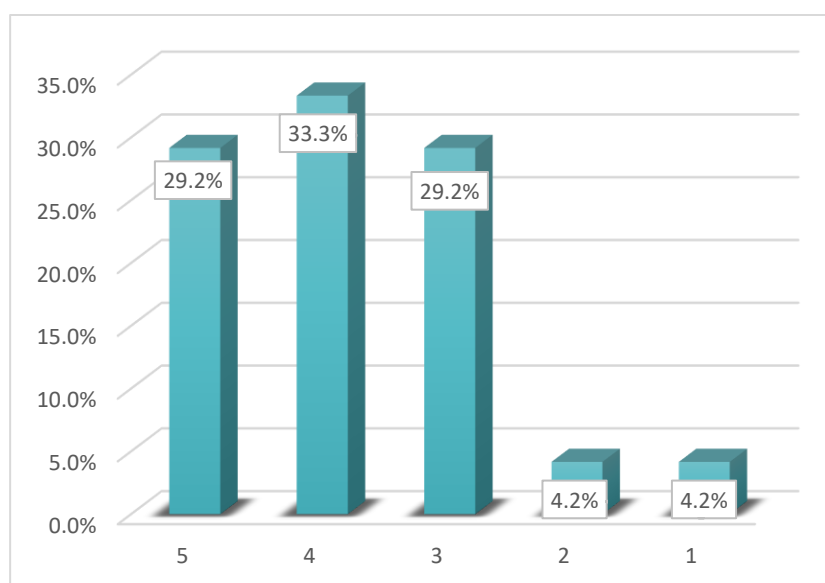
En la Tabla 11 y Gráfico 04, observamos que el índice Muy Alto alcanzó un 37.5%, cuando analizamos el nivel de costo de no conformidad. Un 31.3% se ubicó en el índice Alto y un 20.8%, en el índice Medio. Para finalizar este análisis, un 6.3% de la muestra, indicó un índice Bajo, mientras que un 4.2%, se ubicó en el índice Muy Bajo.

Tabla 12: Flexibilidad - Nivel de respuesta a cambios en los procesos de la organización

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	5	29.2%	29.2%
Alto	5	33.3%	62.5%
Medio	6.5	29.2%	91.7%
Bajo	3.5	4.2%	95.8%
Muy Bajo	4	4.2%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 05: Flexibilidad - Nivel de respuesta a cambios en los procesos de la organización



Fuente: Elaboración propia.

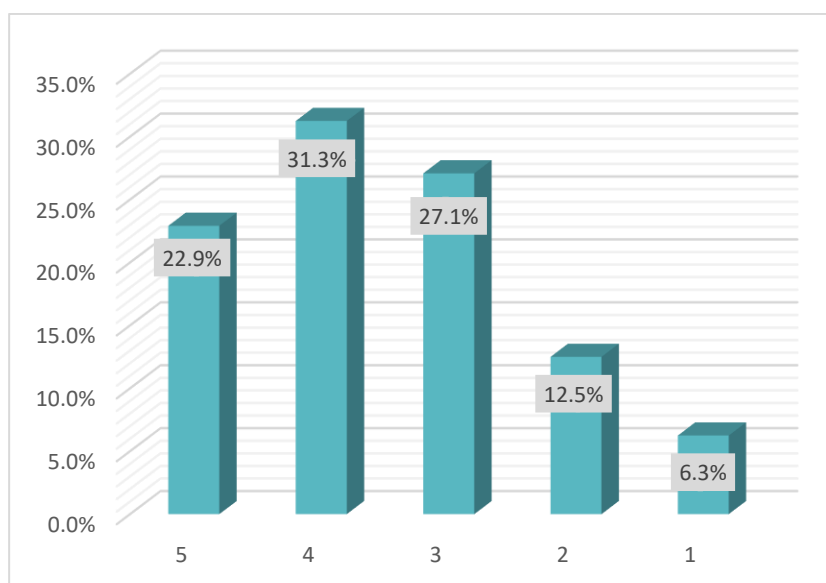
En la Tabla 12 y Gráfico 05, correspondiente al nivel de respuesta a cambios en los procesos de la organización, un 29.2% de los encuestados, se ubicaron en el índice Muy Alto. El índice Alto alcanzó el 33.3% y el índice Medio, 29.2%. Mientras que los índices Bajo y Muy Bajo, obtuvieron, ambos, un 4.2%.

Tabla 13: Flexibilidad - Nivel de adaptabilidad del software a los requisitos cambiantes del proyecto

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	5.5	22.9%	22.9%
Alto	7.5	31.3%	54.2%
Medio	6.5	27.1%	81.3%
Bajo	3	12.5%	93.8%
Muy Bajo	1	6.3%	100.0%
Total	24	100%	

Fuente: Elaboración propia.

Gráfico 06: Flexibilidad - Nivel de adaptabilidad del software a los requisitos cambiantes del proyecto



Fuente: Elaboración propia.

En cuanto al nivel de adaptabilidad del software a los requisitos cambiantes del proyecto, plasmados en la Tabla 13 y Gráfico 06, el 22.9% de la población encuestada (perteneciente a la empresa 'Software y Sistemas del Perú S.A.C' - Iquitos), se ubicó en el índice Muy Alto. El índice Alto, obtuvo un 31.3%, mientras que el índice Medio, representó un 27.1%. Concluye esta parte del estudio, con un 12.5% en el índice Bajo y un 6.3% en el índice Muy Bajo.

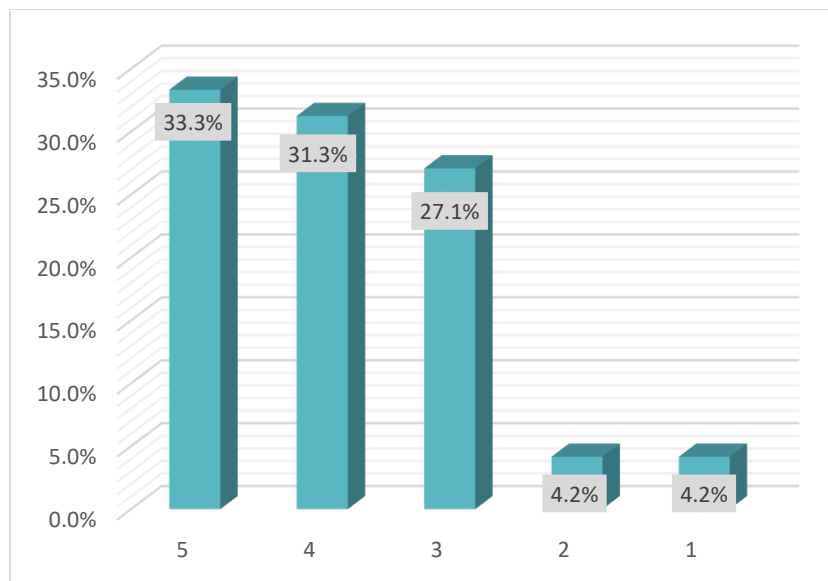
4.1.3 Análisis de los resultados de la variable dependiente – Calidad de Software

Tabla 14: Usabilidad - Nivel de facilidad en el aprendizaje del uso del software

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	5.5	22.9%	22.9%
Alto	7.5	31.3%	54.2%
Medio	6.5	27.1%	81.3%
Bajo	3	12.5%	93.8%
Muy Bajo	1	6.3%	100.0%
Total	24	100%	

Fuente: Elaboración propia.

Gráfico 07: Usabilidad - Nivel de facilidad en el aprendizaje del uso del software



Fuente: Elaboración propia.

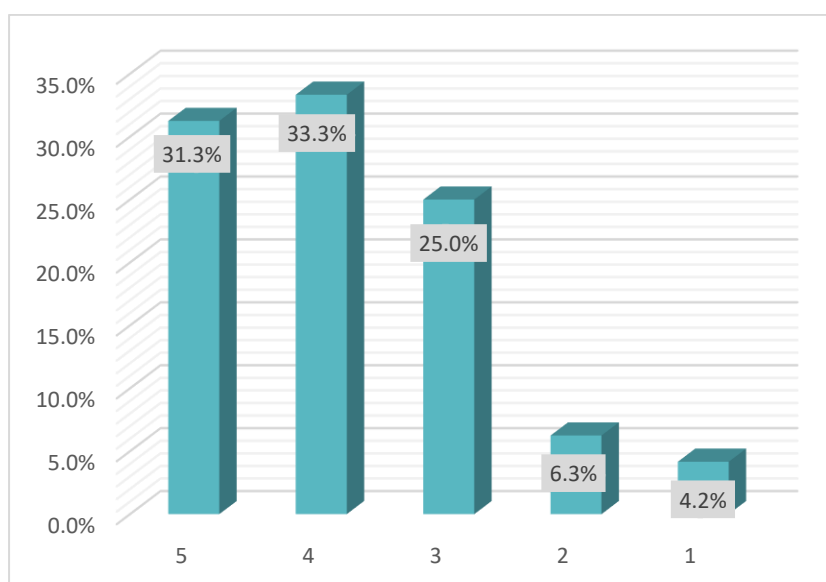
En la Tabla 14 y Gráfico 07, se observa que el 33.3% de los encuestados, indicaron un índice Muy Alto, cuando se evalúa el nivel de facilidad en el aprendizaje del uso del software. Del mismo modo, un 31.3% indicó un índice Alto y un 27.1% indicó un índice Medio. En los dos últimos índices, índice Bajo y Muy Bajo, ambos alcanzaron un resultado de 4.2%.

Tabla 15: Usabilidad - Nivel de facilidad en el uso operativo del software

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	7.5	22.9%	31.3%
Alto	8	31.3%	64.6%
Medio	6	27.1%	89.6%
Bajo	1.5	12.5%	95.8%
Muy Bajo	1	6.3%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 08: Usabilidad - Nivel de facilidad en el uso operativo del software



Fuente: Elaboración propia

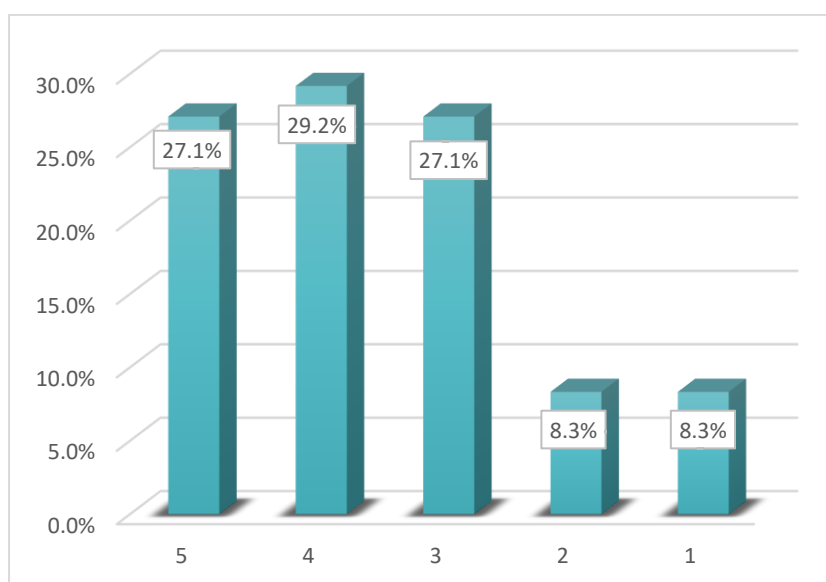
De la revisión de la Tabla 15 y Gráfico 08, se desprende que un 31.3% de los encuestados, indican que el nivel de facilidad en el uso operativo del software, es Muy Alto. Un 32.5%, se obtiene en el índice Alto; mientras que un 25%, lo ubica en el índice Medio. Por último, se registra un 6.3% para el índice Bajo y un 4.2%, para el índice Muy Bajo.

Tabla 16: Fiabilidad - Nivel de tolerancia a fallos del software

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	6.5	27.1%	27.1%
Alto	7	29.2%	56.3%
Medio	6.5	27.1%	83.3%
Bajo	2	8.3%	91.7%
Muy Bajo	2	8.3%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 09: Fiabilidad - Nivel de tolerancia a fallos del software



Fuente: Elaboración propia

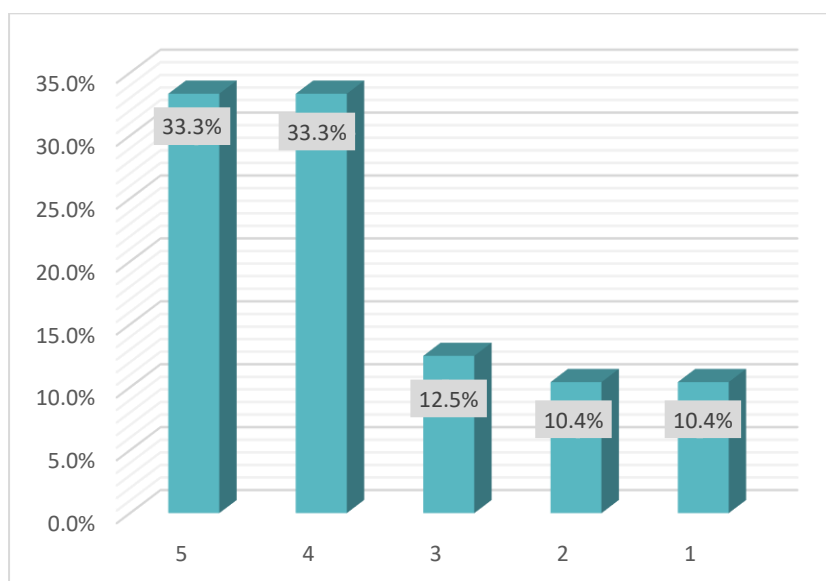
En la Tabla 16 y Gráfico 09, observamos que el 27.1% de los encuestados, indicaron que el nivel de tolerancia a fallos del software, se encuentra en el rango de Muy Alto. En el mismo análisis, un 29.2% de los encuestados, indican un índice Alto y un 27.1%, un índice Medio. Para finalizar este punto, un 8.3% se inclinan por los índices Bajo y Muy Bajo, cada uno.

Tabla 17: Fiabilidad - Nivel de exactitud en el procesamiento de la información

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	8	33.3%	33.3%
Alto	8	33.3%	66.7%
Medio	3	12.5%	79.2%
Bajo	2.5	10.4%	89.6%
Muy Bajo	2.5	10.4%	100.0%
Total	24	100%	

Fuente: Elaboración propia.

Gráfico 10: Fiabilidad - Nivel de exactitud en el procesamiento de la información



Fuente: Elaboración propia.

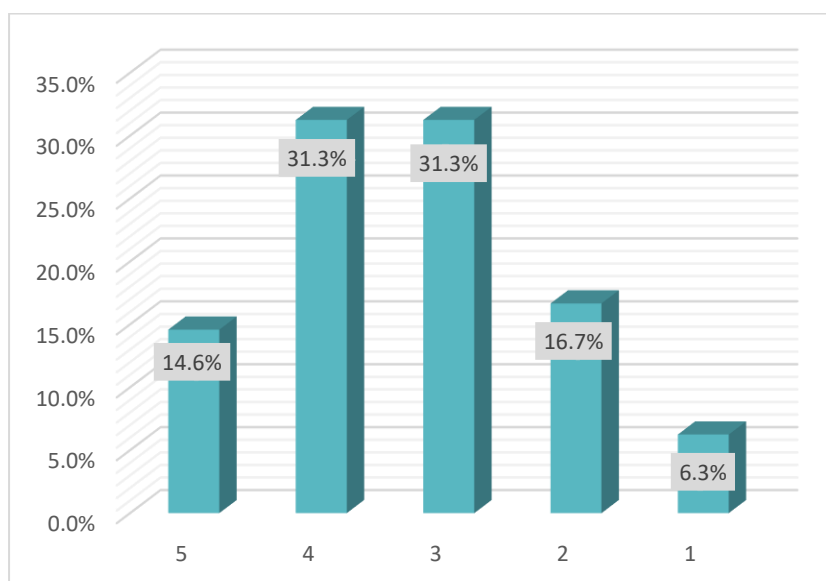
En la Tabla 17 y Gráfico 10, que mide el nivel exactitud en el procesamiento de la información, los índices Muy Alto y Alto alcanzaron, ambos, un 33.3%. Del mismo modo, un 12.5% indica un índice Medio. Cierra esta parte de la investigación, con un 10.4%, en el índice Muy Bajo y Bajo, cada uno.

Tabla 18: Mantenibilidad - Nivel de facilidad para localizar un fallo operativo del software

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	3.5	14.6%	14.6%
Alto	7.5	31.3%	45.8%
Medio	7.5	31.3%	77.1%
Bajo	4	16.7%	93.8%
Muy Bajo	1.5	6.3%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 11: Mantenibilidad - Nivel de facilidad para localizar un fallo operativo del software



Fuente: Elaboración propia.

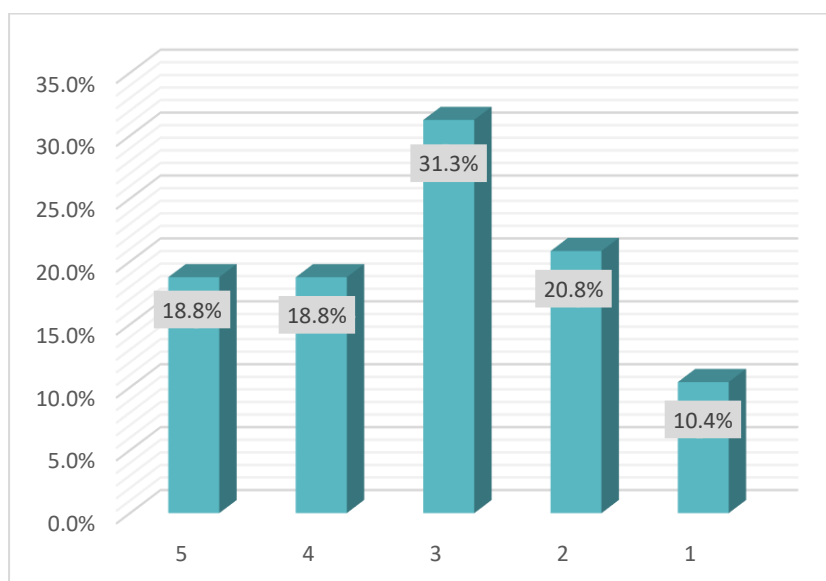
De la revisión de la Tabla 18 y Gráfico 11, observamos que los índices Alto y Medio, alcanzan un 31.3%, mientras que el índice Muy Alto, obtiene un 14.6%. En los índices Bajo y Muy Bajo, los encuestados de la empresa 'Software y Sistemas del Perú S.A.C' – Iquitos, indicaron 16.7% y 6.3%, respectivamente. Esta parte de la investigación, correspondió al nivel de facilidad para localizar un fallo operativo del software.

Tabla 19: Mantenibilidad - Nivel de facilidad para arreglar un fallo operativo del software

Índices	Frecuencia	Porcentaje válido	Porcentaje acumulado
Muy Alto	4	18.8%	18.8%
Alto	4	18.8%	37.5%
Medio	7	31.3%	68.8%
Bajo	5	20.8%	89.6%
Muy Bajo	2.5	10.4%	100.0%
Total	24	100%	

Fuente: Elaboración propia

Gráfico 12: Mantenibilidad - Nivel de facilidad para arreglar un fallo operativo del software



Fuente: Elaboración propia

En cuanto al nivel de facilidad para arreglar un fallo operativo del software, expresados en la Tabla 19 y Gráfico 12, el 18.8% de la población encuestada (perteneciente a la empresa 'Software y Sistemas del Perú S.A.C' – Iquitos), indicó un índice Muy Alto y Alto (para cada uno). El índice Medio, fue el de mayor resultado, con un 31.3%. Concluye esta parte del estudio, con un 20.8% en el índice Bajo y un 10.4% en el índice Muy Bajo.

CAPÍTULO V

DISCUSIÓN, CONCLUSIONES Y RECOMENDACIONES

5.1 DISCUSIÓN

El mundo de la tecnología cambia a una velocidad impresionante y repercute en cada aspecto del accionar humano. Las innovaciones tecnológicas, tienen un gran impacto en el desarrollo de empresas y negocios, que buscan estar a la vanguardia en el mercado, con aplicaciones y sistemas que cubran sus necesidades de funcionamiento.

Estas aplicaciones y sistemas, deben desarrollarse siguiendo una serie de procedimientos, técnicas, herramientas y soporte documental; es decir, metodologías estructuradas, para garantizar su calidad y aprovechamiento eficiente por parte del cliente.

Existe una estrecha relación entre las metodologías de desarrollo de software y la calidad de los mismos, que se puso de manifiesto en la presente investigación.

La población tomada en cuenta para este análisis, representó al 100% de los trabajadores del negocio en cuestión (11 personas), ayudando a la obtención de resultados confiables.

La investigación de (QUEZADA MAURICIO, 2014), titulada “BUENAS PRÁCTICAS DENTRO DEL MARCO DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE, PARA EL USO DE LOS ESTUDIANTES DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”, concluye que la aplicación de una metodología de desarrollo de software, es de gran ayuda para paliar las debilidades encontradas durante el diagnóstico, debilidades en la administración de proyectos, la baja calidad de los productos de software, la falta de automatización en procesos de control y seguimiento, así como en la administración y la integración de módulos de desarrollo de software. En relación a la conclusión anterior, y observando los resultados de la Tabla 11

y Gráfico 04 y, también, de la Tabla 10 y Gráfico 03, notamos que la empresa 'Software y Sistemas del Perú S.A.C' - Iquitos, viene implementando una adecuada gestión en la producción y desarrollo de software, que le permite disminuir riesgos o fallas en el producto final.

En la investigación de (ARAUJO COTACALLAPA, 2017), titulada "SISTEMATIZACIÓN DE LOS PROCESOS DEL PLAN OPERATIVO INSTITUCIONAL UTILIZANDO LA METODOLOGÍA PUDS, PARA LA ELABORACIÓN DEL SISTEMA DE INFORMACIÓN PLAN OPERATIVO INSTITUCIONAL (POIS) EN LA UNIVERSIDAD NACIONAL DEL ALTIPLANO", concluye que, con el Sistema de Información Plan Operativo Institucional, el acceso a la información detallada, es instantánea y está dado por acciones estratégicas. El almacenamiento y la consulta de información en el Plan Operativo Institucional, es eficiente y posible, permitiendo acceder a la información en forma oportuna (tiempo real) para los responsables de los Centros de Costo (Directores de Escuela de Unidad Académica, Jefes de Oficina y Unidad de áreas Administrativas). Así también, se puede realizar el seguimiento y monitoreo de las acciones estratégicas programadas.

En esta línea, el análisis de los datos reflejados en la Tabla 17 y Gráfico 10, expresa que la exactitud en el procesamiento de la información del software, influye en la calidad de las aplicaciones; es decir, el acceso oportuno a información relevante para la construcción de aplicaciones o sistemas, repercute en su calidad, cumpliendo con las expectativas de los clientes de la empresa 'Software y Sistemas del Perú S.A.C' - Iquitos.

(HUAMAN VARAS, y otros, 2017), en su investigación titulada, "DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA MEJORAR LOS PROCESOS DE COMPRAS Y VENTAS EN LA EMPRESA HUMAJU", se comprueba que la metodología Ágil AUP, permitió conocer los riesgos de gestión, en donde estos tengan prioridad en el proceso de desarrollo del sistema, el cual busca darle una mayor satisfacción al cliente, mediante la entrega temprana del software. Se pudo observar que esta metodología, se centra en la funcionalidad que el sistema debe poseer, para satisfacer las necesidades del usuario y siendo

una metodología con claridad en sus actividades, en las etapas del diseño y construcción de software.

Esta conclusión, se relaciona con el análisis obtenido en la Tabla 12 y Gráfico 05, así como en la Tabla 13 y Gráfico 06, que muestran que las estrategias aplicadas en la producción de software, permiten hacer un seguimiento y control de los productos, reduciendo riesgos o imprevistos, elevando la satisfacción del cliente.

(GODOY ALVAREZ, y otros, 2015), en su investigación titulada, “MODELOS DE ACEPTACIÓN DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE”, se concluye que, de los 17 factores que se establecieron como impulsores en la aceptación de las metodologías, los que fueron estudiados más veces en los modelos, son: Compatibilidad, Utilidad, Facilidad de condiciones y Voluntad de uso. Finalmente, los jefes de proyectos de desarrollo, deben tomar en consideración los factores impulsores e inhibidores establecidos en esa investigación, para que el proceso de aceptación de metodologías de desarrollo de software, sea el más adecuado. Esto guarda relación, con la Tabla 09 y Gráfico 02, donde un 56.3% de los trabajadores de la empresa ‘Software y Sistemas del Perú S.A.C’ - Iquitos, indican el uso de metodologías adecuadas, influyen en su desempeño laboral, logrando un trabajo efectivo en la empresa.

Finalmente, (PEZO AREVALO, 2020), en su investigación titulada “IMPLEMENTACIÓN DE SISTEMA DE CONTROL DE ASISTENCIA DOCENTE, PARA MEJORAR LA GESTIÓN DE PERSONAL ACADÉMICO EN LA INSTITUCIÓN EDUCATIVA PÚBLICA DE MENORES N° 60023 - IQUITOS”, concluye que, en relación a su hipótesis general, la implementación del sistema de control de asistencia docente, tuvo efectos de mejora en la gestión de personal académico.

Lo descrito anteriormente, guarda relación con el estudio estadístico reflejado en la Tabla 14 y Gráfico 07 y, también, en la Tabla 15 y Gráfico 08, donde notamos que la facilidad en el uso de software, genera mejoras en la calidad del mismo.

5.2 CONCLUSIONES

Para estar a la vanguardia del desarrollo tecnológico, las empresas requieren aplicaciones y sistemas cada vez más eficientes y especializados. Por ello, el proceso de desarrollo de software, requiere el uso de metodologías que permitan obtener productos de la más alta calidad.

El desarrollo de esta investigación, ha permitido llegar a las siguientes conclusiones:

- a. Existe una relación directamente proporcional entre el adecuado uso de metodologías de desarrollo de software y la alta calidad de las aplicaciones producidas y desarrolladas por 'Software y Sistemas del Perú SAC.' – Iquitos.
- b. La aplicación de metodologías de desarrollo de software, aumentan el grado de eficiencia y productividad laboral, en el personal de la empresa 'Software y Sistemas del Perú SAC.' – Iquitos. El estudio permitió demostrar este beneficio, tanto en acciones comprobables, como en la percepción de sus propios trabajadores.
- c. El uso de procedimientos, herramientas, técnicas y soporte documental, permiten disminuir todo tipo de riesgos, en la funcionabilidad de las aplicaciones producidas por la empresa 'Software y Sistemas del Perú SAC.' – Iquitos. La investigación realizada, comprobó que este uso, influye de manera considerable, en la disminución de costos de mantenimiento de estas aplicaciones, lo que repercute en un óptimo servicio al cliente.
- d. El uso de metodologías de desarrollo de software, permite afrontar de manera eficiente los cambios que se producen en la empresa 'Software y Sistemas del Perú SAC.' – Iquitos, dando respuestas rápidas a contingencias diversas.

5.3 RECOMENDACIONES

Del análisis de la investigación realizada, podemos proponer las siguientes recomendaciones:

- Capacitación constante en la aplicación de las diferentes metodologías de desarrollo de software, al personal encargado de la producción de software, de la empresa 'Software y Sistemas del Perú SAC.' – Iquitos.
- Profundizar el uso de otras metodologías más adecuadas al perfil de la empresa, como la Metodología Lean, con el objetivo de reducir el uso de recursos y mejorar la productividad de la empresa.
- Certificación oficial de los líderes del equipo de desarrollo de software, que garantice un conocimiento profundo en metodologías ágiles.
- Ahondar en el estudio de las metodologías de desarrollo de software y gestión de proyectos, que permitirá tener una visión más amplia, innovadora y centrada en el usuario, de las soluciones ofertadas a los clientes.

REFERENCIAS BIBLIOGRÁFICAS

AgileManifesto.org. 2001. Agile Manifesto. *agilemanifesto.org*. [En línea] 2001. [Citado el: 18 de 10 de 2021.] <https://agilemanifesto.org/iso/es/manifesto.html>.

ALVAREZ, ALONSO, LASA, CARMEN y DE LAS HERAS, RAFAEL. 2017. *Métodos Ágiles y Scrum*. Madrid, España : ANAYA, 2017.

ARAUJO COTACALLAPA, RENÉ LEÓNIDAS. 2017. *SISTEMATIZACIÓN DE LOS PROCESOS DEL PLAN OPERATIVO INSTITUCIONAL UTILIZANDO LA METODOLOGÍA PUDS PARA LA ELABORACIÓN DEL SISTEMA DE INFORMACIÓN PLAN OPERATIVO INSTITUCIONAL (POIS) EN LA UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO 2017*. PUNO : UNIVERSIDAD NACIONAL DEL ALTIPLANO, 2017.

BAUTISTA, LEYDI. 2012. *Control de Calidad de software*. Ica, Perú : s.n., 2012.

BEVAN, NIGEL. 2010. *Los nuevos modelos de ISO para la calidad y la calidad en uso del software*. Madrid, España : Ra-Ma, 2010.

Callejas Cuervo, Mauro, Alarcón Aldana, Andrea Catherine y Álvarez Carreño, Ana María. 2017. *Modelos de Calidad del Software, un estado del arte*. Cali, Colombia : Universidad Libre de Cali, 2017.

COVELLA, GUILLERMO JUAN. 2005. *Medición y evaluación de calidad en uso de aplicaciones web*. La Plata, Argentina : Universidad Nacional de La Plata, 2005.

GODOY ALVAREZ, DIEGO ALONSO y TAYPE CALDERON, ROBERTO SUYOMI. 2015. *MODELOS DE ACEPTACIÓN DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE*. LIMA : UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS, 2015.

GURU, REFACTORING. 2014. REFACTORING GURU. [En línea] 2014. [Citado el: 05 de 11 de 2021.] <https://refactoring.guru/es/design-patterns>.

HEFLO. 2020. HEFLO. [En línea] 01 de 05 de 2020. [Citado el: 03 de 11 de 2021.] <https://www.heflo.com/es/blog/bpm/que-es-bpm/>.

Hernandez Sampieri, Roberto, Fernandez Collado, Carlos y Baptista Lucio, Pilar. 1997. *Metodología de la Investigación*. México : Mc Graw-Hill, 1997.

HUAMAN VARAS, JOSELYN BONNIE y HUAYANCA QUISPE, CARLOS. 2017. *DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA MEJORAR LOS PROCESOS DE COMPRAS Y VENTAS EN LA EMPRESA HUMAJU*. LIMA : UNIVERSIDAD AUTÓNOMA DEL PERÚ, 2017.

IEEE. 1995. *ieeexplore.ieee.org. IEEE Xplore*. [En línea] 1995. [Citado el: 7 de 11 de 2021.] <https://ieeexplore.ieee.org/document/471194>.

MOSZKOWITZ, A. E. 2010. *Modelos de excelencia en la gestión.* Uruguay : IEEM, 2010.

PEZO AREVALO, MARCO ANDRÉS. 2020. *Implementación de sistema de control de asistencia docente para mejorar la gestión de personal académico en la Institución Educativa Pública de Menores N° 60023, Iquitos, año 2020.* IQUITOS : UNIVERSIDAD PRIVADA DE LA SELVA PERUANA, 2020.

PRESSMAN, ROGER. 2010. *Ingeniería del Software. Un enfoque práctico.* Madrid, España : McGraw-Hill, 2010.

PROFILE. 2018. PROFILE.ES. [En línea] 22 de 11 de 2018. [Citado el: 04 de 11 de 2021.] <https://profile.es/blog/principios-solid-desarrollo-software-calidad/>.

QUEZADA MAURICIO, TERESA DOMICILA. 2014. *BUENAS PRÁCTICAS DENTRO DEL MARCO DE UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE PARA EL USO DE LOS ESTUDIANTES DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.* GUATEMALA : UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, 2014.

RODRIGUEZ, MOISES. 2016. *Evaluation of Software Product Functional Suitability: A Case Study.* In: *Software Quality Professional Magazine.* 2016.

ROZO NADER, JANETH. 2014. *Metodología de Desarrollo de Software: MBM (Metodología Basada en Modelos).* Barranquilla, Colombia : Revista Ingeniare, 2014.

Santander Universidades. 2020. Santander Universidades. *Santander.* [En línea] 21 de 12 de 2020. [Citado el: 04 de 11 de 2021.] <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.

SCALONE, FERNANDA. 2006. *Estudio comparativo de los modelos y estándares de calidad del software.* Buenos Aires, Argentina : Universidad Tecnológica Nacional Regional de Buenos Aires, 2006.

TECHLIB. 2018. TECHLIB. [En línea] 2018. [Citado el: 04 de 11 de 2021.] <https://techlib.net/definicion/rup.html>.

Wikipedia. 2020. Wikipedia. [En línea] 2020. [Citado el: 2 de 11 de 2021.] https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software.

Yanina Muradas. 2018. Openwebinars. *openwebinars.net.* [En línea] 08 de 03 de 2018. [Citado el: 18 de 10 de 2020.] <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>.

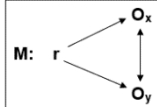
ANEXOS

Anexo 01: Matriz de consistencia.

Anexo 02: Instrumento de recolección de datos.

Anexo 01: Matriz de consistencia

“INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE EN LA CALIDAD DE LAS APLICACIONES PRODUCIDAS POR LA EMPRESA “SOFTWARE Y SISTEMAS DEL PERÚ S.A.C.”, IQUITOS - 2021”

Problema	Objetivo	Hipótesis	Variable	Dimensiones e Indicadores	Índices	Metodología
<p>Problema general:</p> <ul style="list-style-type: none"> ¿Cómo influye el uso de metodologías de desarrollo de software en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021? <p>Problemas específicos:</p> <ul style="list-style-type: none"> ¿De qué manera el uso de metodologías de desarrollo de software promueve el trabajo efectivo en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021? ¿De qué manera uso de metodologías de desarrollo de software permite disminuir los costos de mantenimiento de las aplicaciones por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021? ¿De qué manera uso de metodologías de desarrollo de software permite una rápida respuesta a los cambios en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos - 2021? 	<p>Objetivo general:</p> <ul style="list-style-type: none"> Describir la manera en cómo influye el uso de metodologías de desarrollo de software en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. <p>Objetivos específicos:</p> <ul style="list-style-type: none"> Describir la manera en que el uso de metodologías de desarrollo de software promueve el trabajo efectivo en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. Describir la manera en que el uso de metodologías de desarrollo de software permite disminuir los costos de mantenimiento de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. Describir la manera en que el uso de metodologías de desarrollo de software permite una rápida respuesta a los cambios en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. 	<p>Hipótesis general:</p> <ul style="list-style-type: none"> El uso de metodologías de desarrollo de software influye en la calidad de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. <p>Hipótesis específicas:</p> <ul style="list-style-type: none"> El uso de metodologías de desarrollo de software promueve el trabajo efectivo en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. El uso de metodologías de desarrollo de software permite disminuir los costos de mantenimiento de las aplicaciones producidas por la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. El uso de metodologías de desarrollo de software permite una rápida respuesta a los cambios en la empresa “Software y Sistemas del Perú S.A.C.”, Iquitos – 2021. 	<p>Independiente (X): Metodologías de desarrollo de Software</p>	<p>Efectividad</p> <ul style="list-style-type: none"> Nivel de comunicación entre los interesados del proyecto Nivel de capacidad de auto gestión del equipo de trabajo <p>Costo</p> <ul style="list-style-type: none"> Nivel de costo de conformidad Nivel de costo de no conformidad <p>Flexibilidad</p> <ul style="list-style-type: none"> Nivel de respuesta a cambios en los procesos de la organización. Nivel de adaptabilidad del software a los requisitos cambiantes del proyecto. 	<p>Muy alto, Alto, Medio, Bajo, Muy bajo.</p>	<p>Tipo de investigación: Descriptiva.</p> <p>Diseño de investigación: No experimental transversal de tipo correlacional</p> 
			<p>Dependiente (Y): Calidad de software</p>	<p>Usabilidad</p> <ul style="list-style-type: none"> Nivel de facilidad en el aprendizaje del uso del software Nivel de facilidad en el uso operativo del software <p>Fiabilidad</p> <ul style="list-style-type: none"> Nivel de tolerancia a fallos del software Nivel de exactitud en el procesamiento de la información. <p>Mantenibilidad</p> <ul style="list-style-type: none"> Nivel de facilidad para localizar un fallo operativo del software Nivel de facilidad para arreglar un fallo operativo del software 	<p>Muy alto, Alto, Medio, Bajo, Muy bajo.</p>	<p>Población: todas las personas que laboran en la empresa Software y Sistemas del Perú S.A.C., Iquitos, que en total son 11 individuos</p> <p>Muestra: de tipo aleatoria no probabilística por conveniencia, y estará conformada por 11 individuos.</p>

Anexo 02: Instrumento de recolección de datos

CUESTIONARIO

I. PRESENTACIÓN

El presente cuestionario forma parte del proyecto de investigación titulado: "INFLUENCIA DEL USO DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE EN LA CALIDAD DE LAS APLICACIONES PRODUCIDAS POR LA EMPRESA SOFTWARE Y SISTEMAS DEL PERÚ S.A.C., IQUITOS - 2021".

II. INSTRUCCIONES

Antes de proceder a responder las preguntas del cuestionario debe leer las siguientes instrucciones:

- ✓ Lea cada una de las preguntas y responda de acuerdo con lo que considere pertinente.
- ✓ Debe responder a todas las preguntas del cuestionario.
- ✓ Podrá solicitar aclaración cuando encuentre alguna dificultad en las preguntas.
- ✓ La información proporcionada será de carácter confidencial.
- ✓ No existen preguntas correctas ni incorrectas.
- ✓ Marque con un aspa (X) solamente una de las alternativas para cada pregunta.
- ✓ La duración aproximada para el llenado del cuestionario será de 15 minutos.

Dimensión: Efectividad

Pregunta 01: ¿Cuál considera usted que es el nivel de comunicación entre los interesados del proyecto?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 02: ¿En qué nivel considera usted que la comunicación entre los interesados del proyecto promueve el trabajo efectivo en la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 03: ¿En qué nivel considera usted que se encuentra la capacidad de auto gestión del equipo de trabajo?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 04: ¿En qué nivel considera usted que la capacidad de auto gestión del equipo promueve el trabajo efectivo en la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Dimensión: Costo

Pregunta 05: ¿En qué nivel considera usted que se encuentran en promedio los costos de conformidad de los proyectos de software de la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 06: ¿En qué nivel considera usted que se encuentran en promedio los costos de no conformidad de los proyectos de software de la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 07: ¿En qué nivel considera usted que la adecuada gestión de los costos de conformidad permite disminuir los costos de mantenimiento de las aplicaciones de software producidas en la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 08: ¿En qué nivel considera usted que la adecuada gestión de los costos de no conformidad permite disminuir los costos de mantenimiento de las aplicaciones de software producidas en la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Dimensión: Flexibilidad

Pregunta 09: ¿En qué nivel considera usted que se encuentra la capacidad de respuesta a cambios en los procesos de la organización de la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 10: ¿En qué nivel considera usted que se encuentra la capacidad de adaptación del software producido por la empresa a los requisitos cambiantes del proyecto?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 11: ¿En qué nivel considera usted que la flexibilidad durante la ejecución de un proyecto permite una rápida respuesta a los cambios en la empresa?

Muy alto **Alto** **Medio** **Bajo** **Muy bajo**

Pregunta 12: ¿En qué nivel considera usted que el fortalecimiento de las competencias y habilidades blandas en el equipo de trabajo permite una rápida respuesta a los cambios en la empresa?

Muy alto Alto Medio Bajo Muy bajo

Dimensión: Usabilidad

Pregunta 13: ¿En qué nivel considera usted a la facilidad en el aprendizaje del uso del software?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 14: ¿En qué nivel considera usted a la facilidad en el uso operativo del software?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 15: ¿En qué nivel considera usted que la facilidad en el aprendizaje del uso del software influye en la calidad del mismo?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 16: ¿En qué nivel considera usted que la facilidad en el uso operativo del software influye en la calidad del mismo?

Muy alto Alto Medio Bajo Muy bajo

Dimensión: Fiabilidad

Pregunta 17: ¿En qué nivel considera usted a la tolerancia a fallos del software?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 18: ¿En qué nivel considera usted a la exactitud en el procesamiento de la información del software?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 19: ¿En qué nivel considera usted que la tolerancia a fallos del software influye en la calidad del mismo?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 20: ¿En qué nivel considera usted que la exactitud en el procesamiento de la información del software influye en la calidad del mismo?

Muy alto Alto Medio Bajo Muy bajo

Dimensión: Mantenibilidad

Pregunta 21: ¿En qué nivel considera usted a la facilidad para localizar un fallo operativo del software producido por la empresa?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 22: ¿En qué nivel considera usted a la facilidad para arreglar un fallo operativo del software producido por la empresa?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 23: ¿En qué nivel considera usted que la facilidad para localizar un fallo operativo del software producido por la empresa influye en la calidad del mismo?

Muy alto Alto Medio Bajo Muy bajo

Pregunta 24: ¿En qué nivel considera usted que la facilidad para arreglar un fallo operativo del software producido por la empresa influye en la calidad del mismo?

Muy alto

Alto

Medio

Bajo

Muy bajo